BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PAUL RAYMOND HIGGINBOTTOM, ET AL
Appln. No. 09/441,294
Group Art Unit 2721

AUSTRALIA

OIPE JC76 MAR 07 2000 PATENT & TRADEMARK OFFICE

Patent Office
Canberra

I, KAY WARD, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. PP 7434 for a patent by CANON KABUSHIKI KAISHA filed on 30 November 1998.

WITNESS my hand this
Twenty-second day of November 1999

K. Ward

KAY WARD
TEAM LEADER EXAMINATION
SUPPORT AND SALES

S & F Ref: 442893

ORIGINAL

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION FOR THE INVENTION ENTITLED:

Reconfigurable Sharpening Filter Method and Apparatus

Name and Address of Applicant: Canon Kabushiki Kaisha, incorporated in Japan, of 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 146, JAPAN

Inventors Names: Julie Yan Zhu and Paul Raymond Higginbottom

This invention is best described in the following statement:

Reconfigurable Sharpening Filter Method and Apparatus

The present invention relates to image processing and in particular to a method and apparatus for the detection of predetermined marks or symbols embedded into an image.

In accordance with a first aspect of the present invention there is provided an apparatus for sharpening a digital data comprising:

an array of logic cells, each cell being individually configurable to provide a plurality of functions;

a pipeline register, having a plurality of input means for receiving said digital data and for pipelining said data through said array of logic cells;

a plurality of logic gates for detecting a desired pattern of digital data, and for enabling said logic cells to perform said configured functions on said digital data.

Embodiments of the present invention will now be described with reference to the drawings, in which:

Fig. 1 shows a block diagram of an image reproduction system;

Fig. 2 is a flow-chart of processing steps according to the embodiments of the present invention;

Fig. 3 an example of a mark to be detected in accordance the embodiments of the present invention;

Fig. 4 another example of a mark shown in Fig.3;

Fig. 5 is a block diagram of a pixel preprocessor in accordance with the preferred embodiment of the present invention;

Fig. 6. is a representation of a pixel buffer pipe of the pixel preprocessor of Fig. 5;

Fig. 7 is a flowchart of a colour classification process use in the pixel preprocessor of Fig. 5;

Fig. 8 is a representation of a mark element detection mask according to the preferred embodiment;

Fig. 9 is a representation of a void element detection mask according to the preferred embodiment;

Fig. 10 is a representation of a signature detection mask used in the preferred embodiment of the present invention;

Fig. 11 is a centre portion of the signature detection mask of Fig. 10 in more detail;

Fig. 12 represents the outer and inner ring signature of the sample mark shown in Fig. 3;

Fig. 13 is an overview diagram of a detection apparatus according to the preferred embodiment;

Fig. 14 is a block diagram of the mark detector unit shown in Fig. 13 in more detail;

Fig. 15 is a schematic diagram of a (16-line) line store of the mark detector unit of Fig. 14;

Fig. 16 is a schematic diagram of a mark element detection unit according to the preferred embodiment;

Fig. 17 is a schematic diagram of a (16-line) line store of the mark detector unit of Fig. 14;

Fig. 18A is an example of a sharpening filter mask used to improve a confidence level in a mark detection;

Fig. 18B is a schematic diagram of a sharpening filter implementation of the preferred embodiment of the present invention;

Fig. 18C is a schematic diagram of part of the sharpening filter implementation of Fig. 18B;

Fig. 19 is a schematic diagram of a pattern store shown in the mark detector unit of Fig. 14, in more detail;

Fig. 20 is a zero detect mask in accordance with the preferred embodiment;

Fig. 21 is a schematic diagram of the background detector shown in Fig. 14, in more detail;

Fig. 22 is a schematic diagram of the signature generator shown in Fig. 14, in more detail;

Fig. 23 is a schematic diagram of the signature correlator shown in Fig. 14, in more detail; and

Fig. 24 is a schematic diagram of the compatibility calculator shown in Fig. 14, in more detail.

Overview

Referring to Fig. 1, there is shown an example of a reproduction system 10 upon which the embodiments of the present invention can be implemented. The reproduction system 10 comprising: a scanner device 11 preferably capable of digitally scanning an image to a very high resolution, for example 600 dots per inch (dpi); a computer control system 12; and a printer device 13 for a colour hard copy reproduction of the scanned in image. Examples of such reproduction system include: a colour photocopying machine such as the Canon Colour Laser Copier sold by Canon Inc. of Japan; and a general purpose computer system connected to a scanner device and printer device in the typical manner in which such devices are connected to computer systems. A preferred implementation of the

present embodiments are in hardware incorporated into: input devices, such as image scanners; output devices, such as colour printers; or combined devices having a scanner and printer such as a photocopier.

The image scanned in at the scanner 11 is fed, typically on a pixel by pixel basis, to the computer control system 12 which is able to manipulate the scanned image, for instance, by providing filtering, scaling or transforming functions. The resultant pixel image can be fed to a printer 13, again on a pixel by pixel basis, where it is printed on a medium, such as paper. The scanned image from the scanner 11, normally consists of separate colour channel information for each red, green and blue (R,G,B) colour portions of the scanned image, which is commonly known as an additive colour format. The printer 13 generally prints out the scanned image by a four pass subtractive colour process. The four pass printing process typically consists of separate passes of Cyan, Magenta, Yellow and Black (CMYK) portions of the image. As the printer normally operates on a four pass process, and the image scanner can, typically, on a single scan obtain the RGB colour portions a conversion process is applied to convert from one colour space (RGB) to another (CMYK).

Referring to Fig. 2 there is shown a block diagram representing an overview of the steps processed in accordance with the embodiments of the present invention. An image having thereupon predetermined marking is input, for instance via a digital image scanning device, to a preprocessing step 20. The input image is typically received at the preprocessing step as RGB colour channel information 21 on a pixel by pixel basis where the input image is down-sampled and colour encoded.

Throughout this specification a foreground pixel is to be construed, unless otherwise noted, as a pixel having a pixel (colour) value belonging to a predetermined set of colour values and a background pixel is to be construed, unless otherwise noted, as a pixel not being a foreground pixel.

The preprocessing step 20 substantially provides down sampling and foreground/ background filtering of mark elements to be detected as hereinafter described. One advantage provided by preprocessing is that a predetermined resolution can be passed to a mark element detection step 22 substantially independent of an input resolution and magnification of the input image. Further, by extracting substantially those features of interest from the input image data (e.g. specific colours and background/foreground contrast), an amount of data (bits) required for the encoding of each pixel in relation to these features can be significantly reduced.

Fig.3 and Fig. 4 are two examples of marks that are embedded into an image and detected in accordance with the embodiments of the present invention. The illustrated marks are only for the purpose of the description of the embodiments and should in no way be taken as limiting the scope or spirit of the invention.

Fig. 3 shows a mark 29 comprising a plurality of mark elements 30, represented by square shapes, arranged in a predetermined manner on concentric equilateral triangles 31. The concentric equilateral triangles 31 are not intended to form a visible part of the mark 29 other than to indicate a predetermined arrangement of the plurality of mark elements 30. An outer equilateral triangle 32 substantially represents an outer limit of the mark, that is, an outer boundary 32 for which substantially no part of the mark elements 30 should extend beyond this outer boundary 32. An inner equilateral triangle 33 represents a boundary upon which the mark elements 30 are centred. A further inner equilateral triangle 34 represents a positioning of a mark element 35 in relation to the centre of the mark 29. In addition the mark elements 30 are characterised by a predetermined colour scheme which aids in their identification from the surrounding image colours. Preferably, to improve the confidence level in identifying (matching) the mark elements 30 from surrounding image colours, the predetermined colour scheme for the mark elements 30 and the position or location where the mark 29 is embedded in an image is such that substantially no conflicting colours of the image lie within a predetermined perimeter boundary 36 of each of the mark elements 30. That is, it is desirable not to have colours of the image substantially similar to that of a mark element colour within the predetermined perimeter boundary 36 surrounding a element mark 30.

Fig. 4 shows a mark 39 having an alternate arrangement of mark elements. Both the mark 29 of Fig. 3 and the mark 39 of Fig. 4 are embedded into an image and identified from the image, electronically. An advantage, of embedding a mark in an image and then being able to verify the existence of the mark in the image, is to identify the image and if so desired, to prohibit the reproduction of the image containing such marks. Those skilled in the art will recognise that a multitude of marks can be devised without departing from the scope and spirit of the present invention. For example, the mark elements can be circles of a predetermined size and a mark can consist of a predetermined arrangement of these circles.

Referring again to Fig. 2 the process steps in accordance with the embodiments include the mark element detection step 22 which receives information 21 from the preprocessing step 20 and determines whether or not there are mark elements in the image data received. If a mark element is detected, data representing the coordinate position of the

centre of the mark element, and preferably a strength value representing a statistical confidence level of a degree of matching between the mark element and a predetermined mask used in the detection is recorded. Preferably the data representing the coordinate position of the centre of the mark elements is in the form of a bitmap. The bitmap having, for example, a one (1) where a centre of a mark element is detected and zero (0) otherwise. Since a mark element detection, described hereinafter, is based on a thresholding criteria, each mark element may result in more than one centre being detected for each one of the mark elements 30. Therefore, a sharpening filter step 23 is applied to the bitmap to preferably identify one centre in a small cluster of centres caused by the thresholding criteria. If no marks elements are detected it is assumed that the image does not contain either of the predetermined marks. Next, the sharpened bitmap is further down-sampled 23 before a pattern matching step 24. Therefore, according to the preferred embodiment the image is scanned in at the scanner 11 at 600 dpi, down-sampled at the preprocessing step to 200 dpi and colour encoded. The 200 dpi colour encoded image is converted to a bitmap data substantially representing the coordinate position of the centre of the mark elements at 200 dpi. The resulting bitmap is filtered and further down-sampled 23 to 100 dpi before the pattern matching step 24. The down sampling of the input image and/or resulting bitmap is desirable to reduce errors due to small variations in image colours and/or small variations in a mark element centre detection, respectively. Other advantages of down sampling include a reduction in memory requirements, and improvements in processing efficiencies since less bits of the image are processed at the pattern matching step 24 than would be the case if no down sampling was involved. However the embodiments of the present invention can be practised at various resolutions or without down sampling if it is so desired.

At the pattern matching step 24 a region of the down-sampled 23 bitmap is extracted, preferably including one or more mark element locations (centres detected). A resultant output of the pattern matching step 24 is a "goodness fit" measure for a mark. The goodness fit measure is determined on the basis of the number of mark elements detected in the extracted region and on their relative position or orientation to each other. For example, a goodness fit measure is a percentage value indicating a degree of matching between a known arrangement of mark elements (a known mark) and an arrangement of the detected mark elements (the detected mark).

Preprocessing

Referring now to Fig. 5, there is shown a block diagram of a preprocessing circuit 50 (pixel preprocessor) for implementing the preprocessing step of Fig. 2. The pixel preprocessor 50 comprises: a sub-sample control unit 51; a pixel buffer pipe 52; a pixel selection unit 53; a foreground density checking unit 54; a foreground colour checking unit 55; a colour encoder 56; a line buffer 57; and interface unit 58 referred to hereinafter as a "smart output" unit.

The sub-sample control unit 51 receives an input resolution value and a magnification value via two input lines 60 and 61 respectively. The input resolution value represents the resolution of an image scanned in at the scanner device 11. Preferably, the input resolution value is the vertical resolution value of the scanned image. The magnification value is a scale factor that has been applied to the scanned image to provide the input image data 62 entering the pixel buffer pipe 52. The magnification value represents a reduction or enlargement scaling factor which is typically available on reproduction systems such as colour photocopying machines.

In the horizontal dimension the resolution of an image is fixed by the resolution of an input scan device (e.g. the scanner 11). Typically on a reproduction system such as a colour photocopying machine the magnification of the image in the horizontal direction is performed in software implemented on the photocopying machine and is typically effected at the output stage before printing or reproduction. In the vertical dimension the magnification of the image is typically effected on a photocopy machine in hardware at the input stage.

The sub-sample unit 51 determines a corrected input resolution value (hereinafter "corrected resolution value"), for the vertical dimension and horizontal dimension of the scanned input image, from the input resolution value and the magnification value. For example, for an input resolution value of 600 dpi in each dimension and a magnification value of 200 percent, a corrected resolution value is determined to be 1200 dpi in the vertical dimension and 600 dpi (unchanged) in the horizontal dimension.

Thus, the corrected resolution value reflects the resolution of the input image data 62 received as separate colour channels by the pixel buffer pipe 52.

As previously described, one advantage of preprocessing is that it provides the mark element detection step 22, image data at a predetermined resolution which is substantially independent of the input resolution value and the magnification value of an input image. Thus the pixel preprocessor 50 provides at output a predetermined (desired) resolution

substantially independent of a corrected resolution value for each dimension provided the corrected resolution value is greater than or equal to the predetermined (desired) resolution.

It is assumed that corrected resolution values below the predetermined (desired) resolution value result from a scanned input image that does not require mark detection. An example of such an instant may included, where a scanner device is used to scan an image at a low resolution that upon reproducing the scanned image data on a reproduction device such as a printer, the resulting print image is a poor visual reproduction of the scanned in image.

In a preferred embodiment of the present invention the desired resolution value from the smart output 58 is 200 dpi for each dimension (e.g. 200 x 200 dpi.).

The sub-sample control unit 51 determines an amount of down-sampling required, if necessary, in each of the horizontal and vertical dimensions based on the corrected resolution value and the desired resolution value (200 dpi) for each dimension. A horizontal down-sample signal 64 (H-signal), if required, and a vertical down-sample signal 63 (V-signal), if required, is output from the sub-sample control unit 51 to the pixel buffer pipe 52. The pixel buffer pipe 52 then down-samples the input image data 62 in accordance with the H-signal 64 and/or the V-signal 63 from the sub-sample control unit 51. The desired resolution value of 200 dpi in the vertical dimension and 200 dpi in the horizontal dimension out of the preprocessing circuitry is preferably performed by the pixel preprocessor 50 via an intermediate resolution stage. That is, preferably the pixel buffer pipe 52 instructed by the sub-sample control unit 51 down-samples the input image data 62 to an intermediate resolution, which is further down-sampled to the desired resolution value before final output from the pixel preprocessor 50. In the preferred embodiment the intermediate resolution value is 400 dpi in each dimension, however the intermediate resolution value can be higher or lower than the preferred value of 400 dpi depending upon a desired output resolution value and a predetermined corrected resolution value. If the corrected resolution value in the vertical dimension is greater than the intermediate value of 400 dpi, a V-signal 63 is generated and sent to the pixel buffer pipe 52 to down-sample the vertical dimension of the input image data 62 to the intermediate value. Otherwise, the corrected resolution value in the vertical dimension is substantially equal to or less than the intermediate value of 400 dpi, in which case a V-signal 63 is generated to the pixel buffer pipe 52 to maintain the current resolution of the input image data 62 out of the pixel buffer 52.

The horizontal resolution of the input image data 63 is also down-sampled, in a substantially similar manner to that described with reference to the vertical dimension,

resulting in an intermediate resolution value, of the input image data 62, out of the pixel buffer pipe 52 of 400 dpi.

The horizontal resolution of the input image data 62 is further down-sampled by the pixel selection unit 53 to the desired resolution value of 200 dpi. The down-sampling at the pixel selection unit 53 is performed by selecting **m** pixels out of **n** every pixels, of a current scan line, preferably based on a colour characteristic of the selected pixels.

For example, for each pixel pair on a current scan line, the pixel with the lowest intensity value in the blue channel is, preferably, selected from the pixel pair (i.e. **m**=1 and **n**=2).

Referring to Fig. 6, there is shown a representation 65 of eight (8) consecutive horizontal pixels stored (temporarily) in the pixel buffer pipe 52 of Fig. 5. For the purpose of simplicity and clarity of description, the eight (8) consecutive horizontal pixels of a current scan line are labelled zero (0) to seven (7). An input direction 66 of the current scan line is also shown in Fig. 6. Thus, according to the input direction the first pixel to enter the pixel buffer pipe 52 is labelled pixel seven (7), while the last pixel to enter the buffer pipe 52 is labelled pixel zero (0).

Besides down-sampling, the pixel buffer pipe 52 is used to detect foreground pixel to background pixel transitions and vice versa. Preferably, a foreground pixel is an input image pixel pertaining to a mark element 30 and a background pixel is an input image pixel not pertaining to a mark element 30.

A foreground pixel is distinguished from a background pixel by colour the characteristics of a mark element 30. Thus, a foreground to background pixel transitions or a background to foreground pixel transition on a current scan line represents a mark element intersecting the scan line.

The length of the pixel buffer pipe 52, that is, the number of consecutive horizontal pixels stored in the pixel buffer pipe 52, preferably depends on the width of a mark element outline, measured in pixels, at the operating resolution of the pixel buffer pipe 52 (i.e. the intermediate resolution value - 400 dpi). Accordingly, the representation 65 of Fig. 6 assumes a mark element outline width is, at most, half the length of the pixel buffer pipe 52 (i.e. 4 pixels).

In Fig. 6, pixels labelled zero (0) to three (3) are referred to as "lead pixels" and pixels labelled four (4) to seven (7) are referred to as "lag pixels". Additionally, a current pixel position of the pixel buffer pipe 52 is selected as either pixel positions labelled three (3) or pixel position labelled four (4) depending on a colour characteristic of either pixel. For example, a current pixel processing position of the pixel buffer pipe 52 is selected to be

either pixel position labelled three (3) or pixel position labelled four (4) depending on whichever position has a corresponding pixel with a higher intensity in the blue channel.

A background pixel to foreground pixel transition is detected when a weighted average of intensity values of the lag pixels is greater than a weighted average of intensity values of lead pixels by a first threshold value. Preferably, the intensity value of the lag or lead pixels are selected from a colour channel which predominantly distinguishes each pixel as either foreground or background. The first threshold is, preferably, a predetermined function of the weighted average intensity values of the lag pixels. In a substantially similar manner a foreground to background pixel transition is detected, by the pixel buffer pipe 52, when a weighted average of intensity values of the lead pixels is greater than a weighted average of intensity values of the lag pixels by a second threshold value. The second threshold value is, preferably, a predetermined function of the weighted average of intensity values of the lead pixels.

When a background to foreground pixel transition is detected a foreground transition signal is set high and a counter is initialised to count pixels following the transitions. The foreground transition signal is communicated to the foreground density unit 54, via a separate line not shown in Fig. 5. The foreground transition signal is maintained high until either a foreground to background pixel transition is detected or the counter has counted a predetermined number of pixels, in which case the foreground transition signal is made low. The counter is used as a safety measure in case a foreground to background pixel transition is erroneously not detected, without the counter the foreground transition signal will be maintained high incorrectly, in the event that a foreground to background pixel transition is not detected. The predetermined number of pixels counted by the counter before setting the foreground transition signal to low is preferably the number of pixel in the width of a mark element 30. The counter is cleared upon the foreground transition signal being set low and re-initialized to count pixels upon setting the foreground transition signal high.

Following the pixel selection unit 53, one colour channel 53B, the colour channel which predominantly distinguishes between foreground and background pixels (e.g. blue channel), is input to the foreground density checking unit 54. Remaining colour channels (e.g. Red and Green channels) 53A are input to the foreground colour checking unit 55.

The foreground density checking unit 54 compares an intensity of the single colour channel for each pixel against two threshold values, a background threshold value and a foreground threshold value, and outputs a one (1) or zero (0) for each pixel according to one of the following conditional rules.

The conditional rules for the output of the foreground density checking unit are based on the intensity value of a current pixel. If the intensity of a pixel is:

1) equal to or greater than the foreground threshold value output a one (1) for the pixel;

2) equal to or below the background threshold value then outputs a zero (0) for the pixel; or

3) below the foreground threshold value and above the background threshold value then outputs a one (1) when the foreground transition signal is high and a zero (0) when the foreground transition signal is low.

The binary output of the foreground density checking unit 54 is communicated to the colour encoding unit 56.

The foreground colour checking unit 55 takes as input 53A the remaining two colour channels for the image data output from the pixel selection unit 55 and classifies each pixel of the image data into one of a plurality of allowable colours for a foreground pixel or classifies the pixel colour as "invalid", representing a background pixel. In the preferred embodiment of the invention two bits are assigned to each colour classification, resulting in three allowable colour classifications for a foreground pixel and one invalid colour classification for a background pixel. The binary representation for each colour classification is shown in table 1.

TABLE 1.

Colour	Binary Representation
invalid colour (background)	0 0
colour 1 (foreground)	0 1
colour 2 (foreground)	1 0
colour 3 (foreground)	1 1

Referring to Fig. 7, there is shown a flowchart for the algorithm implemented by the foreground colour checking unit 55 for classifying each pixel of the image data into one of the colour classifications of table 1.

To simplify (clarify) the description of the algorithm of Fig. 7, it is assumed that the colour channel output from the pixel selection unit 52 and input 53B to the foreground density checking unit 54 is the blue channel and the remaining colour channels, namely red

and green, are input 53A to the foreground colour checking unit 55. The input 53A red and green channels are colour corrected 70 to remove possible high frequency effects (AC effects) or artifacts resulting from limitations of the scanning device 11. An example of such artifacts are colour distortions caused by a modulation of spatial frequencies of an input image with a sampling limit (spatial frequency) of the scanning device 11.

Whilst all AC effects are not known a priori, observations of scanned image test patterns have shown that, at least in part, AC effects can be reduced by applying a transformation mapping to the intensity of input image data. An example of such transformation mapping is given by the following equation:

$$F = \frac{F^1 - B \cdot ACF}{1 - ACF}, \qquad \text{(EQ 1)}$$

where F is an intensity value for a foreground pixel of an input image without AC effects, F^1 is an intensity value for a foreground pixel with AC effects, B is the intensity value for a background pixel and ACF is an AC effect factor which has a value between zero and one (i.e.0< ACF <1). Thus, the colour correction step 70 applies a colour correction to the red and green channel of each pixel in accordance with equation 1.

Next, an invalid colour decision step 71 is applied which checks whether or not the red and green channels for a pixel are above desired thresholds. If a result of the invalid colour decision step 71 is true, that is, the red and green channels are above the desired thresholds then the binary code (00) for an invalid colour is output by the foreground colour checking unit 55. Otherwise the decision step 71 returns a false, in which case a second decision step 72 is entered which determines whether or not the red and green channels for the pixel are within a predetermined range of values acceptable as colour 1 from table 1. If the second decision step 72 returns true, that is, the intensity values for the red and green channels of the pixel are within the tolerance specified for colour 1, the foreground colour checking unit 55 outputs the corresponding binary code (01) for the colour 1. However, if the second step 72 returns false, a third decision step 73 is entered. The third decision step 73 determines whether or not the red and green channels are within a predetermined range of values acceptable as colour 2. If the third decision step 73 returns true, the corresponding binary code (10) for colour 2 is output, otherwise the binary code (11) for colour 3 is output by the foreground colour checking unit 55.

Referring back to Fig. 5, the colour encoding unit 56 receives the colour encoded two bit binary output from the foreground colour checking unit 55 and a one bit binary output

from the foreground density checking unit 54 which is used to conditionally toggle switch between the invalid colour (00) and the output of the foreground colour checking unit 55. That is, when the output of the foreground density checking unit 54 is zero (0), then the output of the colour encoder 56 is the binary code (00) for the invalid colour, irrespective of the output of the foreground colour checking unit 55. When the output of the foreground density checking unit 54 is one (1) then the output of the colour encoder 56 is the same as the output from the foreground colour checking unit 55.

As previously described, the resolution in the horizontal dimension out of the pixel selection unit 55 is the desired resolution value (e.g. 200 dpi). However, in the vertical dimension the resolution out of the pixel selection unit 55 can be a value: between the intermediate resolution value (e.g. 400 dpi) and the desired resolution value (e.g. 200 dpi) or equal to either of these resolution values. Whilst the colour coding format (e.g. two bit binary) output from the colour encoding unit 56 is a different colour coding format to that of the input image data (e.g. RGB 8-bits per channel), the output resolution in each dimension from the colour encoding unit 56 is substantially similar to the resolution output from the pixel selection unit 53.

When the vertical resolution output by the colour encoding unit 56 is equal to the desired resolution value (e.g. 200 dpi) then the line buffer unit 57 is disabled and the output from the colour encoding unit 56 is directed through the smart output 58. However, when the vertical resolution is greater than the desired resolution value, (scan) lines of encoded pixel data from the colour encoding unit 56 are buffered through the line buffer 57 in such a manner as to provide an output resolution from the smart output unit 58 at the desired resolution value.

An optional feature of the smart output unit 58 is that of monitoring pixels of a current output scan line and outputing, for each current pixel of the scan line having an invalid colour code (00) value, a corresponding pixel in a previous (adjacent) scan line buffered in the line buffer 57.

For instance, if a current pixel is the fourth pixel of a current output scan line and the pixel has an invalid colour value (00) then the smart output unit 58 outputs for the current pixel a value corresponding to the colour value of the fourth pixel of a previous (adjacent) output scan line. The previous (adjacent) output scan line is buffered in the line buffer unit 57. This optional feature of the smart output unit 58 is only available when the line buffer unit 57 has not been disabled, and is used to stabilise the output of the pixel preprocessor 50 against unwanted artifacts resulting from fine horizontal lines in an input image (if any are present).

Mark Element Detection

In the preferred embodiment of the present invention the mark elements are detected using a binary morphological filtering approach. Morphological filters extract image components that are useful in representing and describing region shapes. Such image components include boundaries, skeletons and convex hulls to name a few. The preprocessing of the input image, as previously described, yields a coded representation that discriminates between foreground and background pixels. Based on this coding representation, a mark element detection step 22 can examine many more pixels of the input image data for a fixed memory requirement, since each pixel is represented by one or two bits.

Referring to Fig. 8, there is shown an element detection mask 80 for detecting the mark elements 30 of Fig. 3 or Fig. 4. The element detection mask 80 is a window of 16x16 cell locations 81 which correspond to pixel locations of a 16x16 array of pixels from the coded representation output from the preprocessing step 20. That is, the element detection mask 80 represents the pixel locations of a 16x16 portion of the sub-sampled image resulting form the preprocessing step 20. The dimensions of a mask (or window) is typically determined by features such as, processing capabilities, the size in pixels of a mark element to be detected and a spacing between mark elements. A convenient size for the element detection mask 80 of the present embodiment is a 16x16 cell matrix.

Predetermined cell locations of the element detection mask 80 shown in Fig. 8 are labelled with the letter "X" 82. The x-labelled cells 82 indicate cell locations which are to substantially coincide with foreground pixels for a mark element to be detected.

Fig. 9 shows a second mask 90, referred to hereinafter as the "void element mask". The void element mask 90 is substantially similar to the element detection mask 80, excepting that predetermined cell locations are labelled with the letter "Y" and the y-labelled cell locations do not correspond to the cell locations 82 of the element detection mask 80 labelled "x". That is, when the void element mask 90 is overlaid upon the element detection mask 80 at corresponding cell location there is substantially no overlap of the y-labelled cells with the x-labelled cells of the masks 90 and 80 respectively.

A characteristic feature of each mark element 30 is that within a predetermined perimeter boundary 36 of each element there is substantially no foreground pixels other than those foreground pixels pertaining to the mark element 30 itself. The void element mask 90 is used to provide a degree of confidence in detecting a mark element 30 by checking for this characteristic feature. That is, the void element mask 90 is used to

determined whether or not pixels surrounding a mark element and preferably at the centre portion of the mark element are substantially background pixels.

The element detection mask 80 and the void element mask 90 are used together to detect mark elements 30 embedded an input image. Both masks 80 and 90 are applied to a portion of the encoded image resulting from the preprocessing step 20. On application of the element detection mask 80 to a portion of the encoded image a first score is determined and a second score value is determined for an application of the void element mask 90 on the same portion of the encoded image. The first and second score values are thresholded against a first and second threshold value respectively. If both score values exceed their corresponding threshold value a one (1) is recorded, otherwise a zero (0) is recorded. The masks 80 and 90 are translated to another portion of the encoded image and the process of determining the score values, thresholding the score values and recording a one (1) or zero (0) is repeated.

The mark element detection step 22 proceeds until both masks 80 and 90 have processed each pixel of the encoded image resulting in a binary bitmap comprising: a one (1) at substantially each position of the bitmap representing a centre location of a mark element; and zero (0) otherwise.

For rotationally invariant mark elements (e.g. circular mark elements) only a single pass, on each pixel, of the mark element detection step is required to produce a bitmap representation of centre location of the mark elements. However, for mark element not rotationally invariant, several passes of the mark element detection step 22 are required to account for a possibility of one of a plurality of rotation angles for which an image may be scanned in at the scanner device 11. For each of the several passes a modified element detection mask and a modified void element mask is applied.

The modified element detection mask and the modified void element mask for each of the several passes is determined from a specified rotation of the element detection mask 80 and the void element mask 90 respectively. For example, in detecting a square mark element 30 (Fig. 3 or Fig. 4) an effective number of (modified) element detection masks and (modified) void element masks is 16, one for each 6 degree rotation from 0 degrees to 90 degrees inclusive. Bitmaps produced from each of the several passes of the mark element detection step 22 (i.e. one pass for each 6-degree rotation) are combined to produce a single bitmap representation which is sent to the sharpening and down-sampling step 24.

The mark element detection step 22 of the preferred embodiment is hereinbefore described with reference to two detection masks 80, 90, however, the mark element

detection can be performed with a single mask combining the attributes of the mark element detection mask 80 and the void element mask 90. That is, in practise, the element detection mask 80 and the void element mask 90 for detecting a mark element, can be implemented in parallel without departing from the scope and/or spirit of the present invention.

Returning to Fig. 2, the bitmap representing centre location of mark elements is preferably sharpened with a sharpening filter and further down-sampled 23 to reduce possible ambiguities in the centre locations introduced by the mark element detection step 22. The down-sampled sharpened image is processed by a then pattern matching step 24 which determines whether or not the mark elements are in a correct pattern arrangement in accordance with the mark to be detected (e.g. Fig. 3 and/or Fig. 4). The pattern matching step 24 provides a measure indicating a "goodness of fit" between a detected mark and a predetermined "known" mark and communicates the measure value to an evidence combiner step (not shown in Fig. 2) which determines whether to enable or disable specific features such as copying on a photocopier.

Pattern Matching

Once centre locations of the mark elements 30 have been detected and represented on a bitmap, an arrangement of the centre location is checked against known arrangements (marks of Fig. 3 and of Fig. 4) and a confidence value or a "goodness of fit" measures is determined.

Referring to Fig. 10, there is shown a signature detect mask 1000 comprising 32x32 matrix of cells 1011 which is used by the pattern matching step 24 for determining an outer ring signature for an arrangement (or pattern) of mark element centres. In addition Fig. 10 shows a circular region 1002 divided into 36 ten degree sectors labelled 0 to 35. The signature detect mask 1000 represents a 32x32 pixel region of the sharpened, down-sampled bitmap resulting from the mark element detection step 22.

A predetermined number of matrix cells (shown shaded) arranged in a circular ring pattern 1003 are used to determine an outer ring signature for each mark (Fig. 3 or Fig. 4). Overlaid upon the signature detect mask 1000 of Fig. 10 there is also shown the outer mark elements 30 of the mark pattern arrangement of Fig. 3. A centre location of the outer mark elements of the pattern arrangement of Fig. 3 are shown coincident with three mask cells, on the circular ring pattern 1003, one 1004 at sector 0, one 1005 at sector 17 and one 1006 at sector 29.

Referring to Fig. 11, there is shown a central 10x10 cell portion 1007 of the signature detect mask 1000 of Fig. 10 in more detail. The central portion 1007 is used to determine an inner ring signature for a mark, indicating a position of a centre mark element relative to the outer mark elements 30 of a mark. Fig. 11 shows a circular portion 1100 preferably divided into 18 sector 1101, each sector spanning a 20-degree angle about the centre of the signature detect mask 1000 of Fig. 10. The angle spanned by a sector of the central portion 1007 is greater than the angle spanned by a sector of the circular region 1002 so as to increase the sensitivity of the angular resolution (e.g. pixels per degree). Also, shown in Fig. 11 is a centre location 1102 of the central mark element shown in the mark pattern arrangement of Fig. 3.

By way of example, Fig. 12 shows an outer ring signature 1200 and an inner ring signature 1201 for the mark shown in Fig. 3. A first row 1202 of the table 1203 in Fig. 12 represents the labels of each sector shown in Fig. 10. A second row represents the outer ring signature 1200 for the outer mark elements 30 and a third row of the table 1203 represents an inner ring signature of the central mark element of Fig. 3.

The outer 1200 and inner 1201 ring signatures consists of a plurality of error measures (or distance measures) values which indicate how close a sector is to a sector containing a mark element centre. For example, at one extreme a zero (0) error measure value indicates that a mark element centre location is within the corresponding sector and at another extreme an error measure value of three (3) indicates that the corresponding sector is far removed from a sector containing a mark element centre location. Between these two extreme values are error measure values one (1) and two (2) which indicate that the corresponding sector is distant (removed) from a sector containing a mark element centre location but not quite as much as that indicated by the error measure value three (3). The error measure value can be mapped to values other than the values 0 to 3 without departing from the scope and spirit of the present invention.

The signature detect mask 1000 (Fig. 10) is applied to an encoded image one pixel at a time and an outer and inner signature is determined at each application. The signatures determined on each application are compared with a plurality of expected "known" signatures for each mark represented in Fig. 3 and Fig. 4. The plurality of expected "known" signatures comprise a signature for each possible (10-degree) rotation of each mark of Fig. 3 and Fig. 4.

At each comparison between the determined signatures and the known signatures a confidence level or "goodness of fit" measure is determined. The confidence level or "goodness of fit" measure is thresholded by an evidence combiner to determine whether or

not an input device (e.g. scanner) or an output device (e.g. printer) or a combination of such devices (e.g. photocopier) will enable a copying of an input image.

Referring to now Fig. 13, there is shown a block diagram of a mark detection apparatus 1300 according to the preferred embodiment of the present invention. The mark detection apparatus 1300 comprising: the pixel preprocessor 50 (Fig.5); a mark detector 1301 for performing the mark element detection 22, the down-sampling and sharpening 23 and the pattern matching step 24 of Fig.2; and an evidence combiner module 1302. The mark detector 1301 has an internal control bus 1303 accessible to an external microprocessor and includes substantially all the necessary memory and logic to detect marks, preferably in real time.

The mark detector takes its input from the Pixel Preprocessor 50 and provides detection probability data to the evidence combiner module 1302. The evidence combiner module 1302 from the detection probability data (i.e. goodness of fit measure) determines whether the photocopier is to allow an image to be copied. A microprocessor interface allows the mark detector's RAM and registers to be accessed from an external microprocessor.

The mark detector uses both a photocopier input clock, *pclk* and a high frequency synthesised clock, *hclk*. The photocopier input clock *pclk* is used at the front end where pixel rate processing is possible, while the synthesised clock *hclk* is needed in a pattern matching detector section described hereinafter to improve processing throughput. Table 2. lists some of the clocks and global signal definitions.

Table 2.

Name	Type	Definition
pclk	input	Photocopier input clock (1 - 25 MHz)
hclk	input	Derived high frequency clock (2 - 200 MHz)
prst	input	Global reset synchronized to *pclk*
hrst	input	Global reset synchronized to *hclk*

The global reset signal *hrst* is synchronous to synthesised clock *hclk* and global reset signal *prst* is synchronous to photocopier input clock *pclk*. Both resets should clear all state machines and registers in the circuits clocked by the respective clocks.

Registers and memory data areas within the mark detector 1302 are able to be read and written via the internal control bus from the external microprocessor. In addition they can be written by a data supply unit.

A pixel bus (PBus) conveys pixel information in real time from the photocopier via the pixel preprocessor to the mark detector 1301 and an evidence combiner bus (EBus) conveys evidence information from the mark detector 1301 to the evidence combiner module 1302.

The mark detector module 1301 is shown in more detail in Fig. 14 and is where each mark (Fig 3 and Fig 4) is detected. The mark detector 1301 comprises two line stores 1400,1401, a pattern store 1402 and logic required to scan an incoming pixel stream in real time and detect both mark elements and marks (i.e. entire mark element pattern arrangement) and report detection probabilities to the evidence combiner module 1302.

The preprocessed pixel data is written into the 16-line store 1400 in real time and read out in lock step with the input data stream. A mark element detector 1403 scans the pixel data in the 16-line store, looking for the mark elements 30 of Fig.3 and Fig.4. Once located, mark element centres are stored in the pattern store memory after down-sampling to a predetermined resolution (preferably 100 dpi). Enough lines of down sampled pixel data are stored, in the pattern store 1402, to hold a mark at any (allowable) rotation. An arrangement detector 1404 (comprising background detector 1405, signature generator 1406 and signature correlator 1407) looks for the two possible marks (Fig. 3 and Fig. 4) and determines a correlation error. A compatibility calculator converts the correlation error into a compatibility percentage value (goodness of fit measure) which is passed to the evidence combiner 1302.

Turning now to Fig. 15 there is shown a more detailed diagram of the 16-line store 1400 of Fig 14. A random access memory (RAM) array 1500 stores fifteen complete lines of pixel (video) data at 200 dpi with just 2 bits per pixel resulting from the pixel processor 50. This comprises 2500 words of 30 bits per word. The RAM array 1500 is periodically executing read-modify-write cycles with a 2-bit rotation. New, incoming 2-bit data is merged with the bottom 28 bits of the data read from the RAM at the same address before being written back into the RAM 1500. The full 30-bit word read out of the RAM 1500 is passed on to the mark element detector block 1403 along with the two incoming bits from the pixel preprocessor 50.

A 12-bit pixel counter 1501 tracks addresses within the RAM array corresponding to pixel position along a line of pixel data. Writing into the RAM is enabled during the active line time. The counter 1501 is enabled at the same time and is reset at the start of each line.

A first comparator 1502 is used to assert a *h_start* signal which indicates the horizontal start of when element detection should begin in the horizontal direction.

A line counter 1503 is reset at the start of each page and is incremented at the start of each line. A second comparator 1504 is used to generate a *v_start* signal which is used to indicate the start of when element detection should commence in the vertical direction.

Returning to Fig. 14, the mark element detector 1403 operates on the 32-bit data stream from the 16-line store 1400 in chunks of 16 words at a time. Conceptually, it stores a 16 x 16 array of pixels (2 bits per pixel) and applies each of two masks 80,90 to the image data. As previously described in the Mark Element Detection section, one mask, the element detection mask 80, is used to detect foreground (coloured) pixels and another mask, the void element mask 90, is used to detect background (coloured) pixels.

Each cell position of the element detection mask 80 label with an 'X' contributes to the detection calculation. For each of these cells, if a foreground pixel is detected, one is added to a total count of foreground hits. Typically, for marks having mark elements of more than one colour a total count of foreground hits for each colour is recorded. If the total number of foreground hits (per colour) exceeds a threshold value, then a foreground element is detected.

Again as previously described the void element mask 90 is used to detect background (coloured) pixels and each cell position labelled with a 'Y' contributes to the detection calculation. For each of these cells, if a foreground pixel is detected, one is added to a total count of background hits per complementary colour. Again for each mark element colour there can be a complementary background colour which is not in conflict, for the purpose of detection, with the mark element colour. If the total number of background hits (per colour) exceeds a threshold value, then a background element is detected. Cells labelled with a 'Y' at the centre of the void element mask 90 (Fig. 9) are preferably detected separately to the peripheral Y-labelled cell of the void element mask 90.

A mark element 30 is considered detected when both the foreground element and the background element is detected upon the same portion of image data.

A portion of the mark element detector 1403 of Fig. 14 is illustrated in more detail in Fig. 16. Fig. 16 shows the portion of the detector 1403 for a single foreground colour and a single rotational orientation. Three such portion are required to detect mark elements having three foreground colours (shown in table 1) and 3n times as many portion of the mark element detector for the three foreground colours and n (possible) rotational orientations of the mark elements.

A 32-bit data stream 1600 from the 16-line store 1400 is compared to the foreground colour in 16 (2-bit) equality comparators 1601. The result, of the comparators 1601, is passed into three different adder trees 1602, 1603 and 1604, which count the number of foreground pixels, background pixels within each of 16 vertical slices through the image. The 16 vertical slices represent the 16 columns, of cell locations, of either the element detection mask 80 or the void element mask 90. Rather than store all 16 vertical slices of a mask at once, one vertical slice at a time is processed and 16 partial sums corresponding to the 16 vertical slices are stored.

The foreground (FG) adder trees 1602 is hard-wired to look for foreground pixels in cell locations labelled 'X' of the element detection mask 80, while the background (BG) adder tree 1603 and the centre adder tree 1604 looks for background pixels in cell locations labelled 'Y' of the void element mask 90. The centre adder tree 1604 is hard-wired to look for background pixels in cell locations (labelled 'Y') at the centre of the void element mask 90, while the BG adder tree 1603 checks for peripherally located Y-labelled cells of the mask 90.

The FG adder tree 1602 produces 2 partial sums which are added into a foreground (FG) accumulator network 1605. The FG accumulator network 1605 is a shift register which stores the result of adding the incoming partial sum from the FG adder tree 1602 to a previous stage of the FG accumulator network and passes it on to a next stage. The number of bits required at each stage thus grows logarithmically to a maximum of 6 bits. For foreground pixels, only the middle 8 slices actually contribute pixels to the total, but a further 4 pipeline stages are required to line up the data with the result of a background (BG) accumulator network 1606. A result of the last stage of the FG accumulator network 1605 is compared 1610 to a foreground threshold value (fg_threshold), stored in a configuration register, to determine whether a hit is detected for the foreground pixels of this colour. The result of the comparison is then pipe-lined in a four-stage shift register 1607 to line up the result with the BG accumulator network 1606.

The BG adder tree 1603 produces 2 partial sums which are added into the BG accumulator network 1606. In a substantially similar manner to the FG accumulator network 1605, but requiring different numbers of bits at each stage, the BG accumulator network 1606 accumulates the partial sums. The number of bits required grows to a maximum of 7 bits for this accumulator and the last stage of the BG accumulator network 1606 is compared 1611 to a background threshold value (bg_threshold), stored in a configuration register, to determine whether a hit is detected for the background pixels of this colour.

The centre adder tree 1604 produces a single partial sum which is added into a centre accumulator network 1608. Again, this is similar in operation to the accumulator networks already described, excepting that it has a different number of bits at each stage, to a maximum of 3 bits. The last stage of the centre accumulator network 1608 is compared 1612 to a centre threshold value (cntr_threshold), stored in a configuration register, to determine whether a hit is detected for the centre pixels of the present colour. The result of the comparison is then pipe-lined in a seven-stage shift register 1609 to line up the result with the BG accumulator network 1606.

Finally, the result of all three comparisons are ANDed together to produce the *colx_hit* signal which indicates that a mark element of the present colour 1613 has been detected. Upon each mark element detection, a centre location of the detected mark element is passed to the 5-line store 1401.

The 5-line store 1401 is illustrated in more detail in Fig. 17 and is used to hold, temporarily, mark element centre (location) data at 200 dpi while a sharpening operation is performed on the data. The store 1401 capacity is 2500 words by 5 bits in organisation (12.5 k bits). An *sf_out* signal from a Sharpening Filter 1408 (Fig. 14) is written into a RAM array 1700, 5 clock cycles after the signal is read out of the RAM array 1700 into the Sharpening Filter 1408 via *sf_in* signal 5 bit wide data bus 1701. A write address is preferably five RAM memory addresses values below a current read address value to account for a 5-stage pipeline delay in the Sharpening Filter 1408. For this reason two addresses value are kept, one for reading and one for writing to the RAM 1700.

A read address counter 1702 is 12 bits wide and is reset by an inverted *r_start* 1703 signal from the 16-line store 1400 and enabled by a *pb_valid* 1704 signal from the pixel processor module 50.

A write address counter 1705 is also 12 bits wide and is reset by an inverted *h_start* 1706 and enabled by the *pb_valid* 1704 signal from the pixel processor module 50.

The *r_start* 1703 signal is asserted high 5 pixels before *h_start* 1706 signal is asserted high. This ensures that a read address value is always 5 greater than a write address value.

A multiplexer 1707 drives read addresses or write addresses to the RAM array 1700 according to the *pb_valid* 1704 signal value.

A 3-bit counter 1708 and encoder 1708 count valid lines from the start of the image and turns on the line enables, *l_valid* 1710 signal, one at a time as each valid line is received from the bitmap image resulting from the pixel processor module 50. At the end of the bitmap image it turns them off again. This ensures that no elements can be detected

outside the image area and that the sharpening filter is correctly applied at the image boundaries.

Two one-bit counters 1711 and 1712 are used to generate a *p_wen* 1713 signal for the pattern store 1402 and an *odd_line* 1714 signal for the arrangement detector 1404. The *p_wen* 1713 signal is asserted on a second valid pixel in every pair of valid pixels, on every second valid line (the odd lines) of pixels.

Turning now to Fig. 18A, a Sharpening Filter algorithm requires the application of the sharpening filter mask 1800 (or matrix). A typical mask for this purpose is shown in Fig. 18A and process a 5x5 pixel region of a bitmap image upon each application to produce a single pixel output value. For each 5x5 area of an image, if the centre location is coincident with a bitmap pixel value of '1' and if each location labelled with a '0' in the filter mask 1800 is coincident with a bitmap pixel value of zero, then all bitmap pixels, coincident with the positions labelled 'X' on the filter mask 1800, are set to zero.

Applying this sharpening filter to the output of the mark element detector 1403, has the effect of removing any centre locations that are close to a given centre location in a small group of up to four centre locations without affecting centre location spaced by at least one non-centre pixel position. This helps reduce multiple occurrences of the same centre in the pattern store 1402.

The sharpening filter of the preferred embodiment advantageously provides a degree of flexibility in filter configurations. Each cell in the sharpening filter mask 1800 can be individually programmed to have one of four possible functions, depending on its 2-bit configuration field. A configuration value of '00' means that the cell will have no effect on the pattern match and the cell will not be cleared when a pattern match is found. A value of '01' means that the cell will be cleared when a pattern match is found. A value of '10' or '11' means that a value of '0' or '1' will contribute to the pattern match respectively (the cell must match '0' or '1' respectively for a pattern match to be found).

Preferably, the sharpening filter 1408 also down-samples the data in the 5 line store 1400 to 100 dpi before the filtered data is passed on to the pattern store 1402.

Referring now to Fig 18B, there is shown a block diagram of an implementation of the sharpening filter of the preferred embodiment. A 5 x 5 array of flip-flops 1801 that stores 5 vertical slices through the 5 Line Store 1401. The flip-flop array forms a pipeline register that shifts in the direction indicated by the arrow 1802 of Fig 18B. A plurality of gates 1803 perform a pattern matching function that causes selected flip-flops 1801 to be cleared via corresponding set of logic cells 1804 marked 'D' in Fig 18B. These logic cells 1804 also generate detect signals used to do the pattern matching. Data read out of the final

pipeline stage 1805 (i.e. sf_out(0) to sf_out(4)) is written into the 5 Line Store 1401 while data read out of the 5 Line Store 1401 is loaded into the pipeline at inputs (i.e. sf_in(0) to sf_in(4)) which connect to the output 1701 of the 5 Line Store 1401. Incoming data can be cleared to background by negation of the *l_valid* 1710 signals (i.e.. l_val(0) to l_val(4)) from the 5 Line Store 1401.

Colour hit signals from the PC Element Detector are ORed together to be written into the 5 Line Store. If any of the three colour hits is asserted, then an element centre has been detected and is stored in the 5 Line Store as a '1' on the first line (bit 4).

The 5 Line Store data is thus being continuously shifted in the direction of the arrow1802 shown and top to bottom of Fig 18B with the incoming data stream so that it behaves like a 5 line window into the original image.

The bottom two pixels out of the end of the pipeline are ORed with 1 cycle delayed versions of the same to generate a `ps_in` signal. This is the maximum down-sampled (100 dpi) data which is subsequently stored in the Pattern Store 1402.

Referring now to Fig 18C, there is shown in more detail the configuration of the logic cells 1804 of Fig 18B. A `conf0` signal 1810 and `conf1` signal 1811 are used to configure the sharpening filter and represent a first bit and a second bit signal of the configuration value as previously described. That is a configuration value of '00' represents `conf0`= 0 and `conf1`= 0.

The Pattern Store 1402 shown in Fig. 14 is illustrated in more detail in Fig. 19 and is substantially similar, in operation, to the 16-line store 1400, excepting only 1250 words of, RAM memory 1900, are needed because of the reduced resolution. Preferably, the Pattern Store 1402 stores 30 lines of data to enable an efficient application of the masks of Fig. 10 and Fig. 11 by the arrangement detector 1404.

Referring to Fig 20 there is shown a zero detect mask 2000 which when overlaid upon the content of the pattern store 1402 is used to detect whether or not a mark element centre location in coincident with an X-labelled cell of the mask 2000. A Zero Detect mask, which has a substantially similar function on a mark as the void element mask 90 has on a mark element, is used on the content of the pattern store 1402 to trigger the signature generator of the arrangement detector 1404. The Zero Detect mask is apply by the background (BG) detector 1405 and if a false results, that is, there is a mark element centre location coincident with an X-labelled cell of the mask 2000, then the mask is move to another pixel in the pattern store and re-applied. Otherwise a true is returned by the BG detector 1405, that is, there are no mark element centre location where there should not be,

the signature detector is triggered to determined a signature for a current content of the pattern store where the mask 2000 is applied.

The Background Detector 1405 of Fig 14. is illustrated in detail in Fig. 21. Data from the pattern store 1402 enters a Background OR-Tree 2100 where vertical slices through the pattern store 1402 are ORed together to produce the 15 partial terms. Only 15 are needed because the zero detect mask 2000 has reflective symmetry about a vertical slice through the middle of the mask 2000, so the 15 partial terms are repeated in reverse order to make up 30 terms of a background (BG) OR-shift register 2101. Each of the resultant 30 terms is ORed with the output of a previous stage of the BG OR-shift register 2101 and stored back into a current stage of the register 2101. A single bit stream data output from the BG OR-shift register 2101 represents an OR result of all pixels coincident with the X-labelled of the zero detect mask 2000. If any bit is asserted (indicating the presence of an element) then the *template_en* 2102 signal is de-asserted, indicating that a valid mark cannot be flagged at this location.

The background detector 1405 also determined whether or not there are a correct number of mark element centre location to determine an outer ring signature and inner ring signature in accordance with the signature detect mask 1000 of Fig 10 and Fig. 11 respectively. To this end, an *outer_sum* 2103 data signal is compared to the values in the *outer_lo_threshold* 2104 signal and *outer_hi_threshold* 2105 signal registers in a pair of 3-bit comparators 2106 and 2107. If both compares, from comparators 2106 and 2107, are true then the required number of mark element centre locations have been detected for determining a correct outer ring signature. A substantially similar comparison is done for an *inner_sum* 2108 signal using values in a *inner_lo_threshold* 2109 and an *inner_hi_threshold* 2110 registers in a pair of 2-bit comparators 2111 and 2112. If both compares, from comparators 2111 and 2112, are true then the required number of mark element centre locations have been detected for determining a correct inner ring signature. If both pairs of comparisons are true and the shift register 2101 output is false, then a valid mark detect is possible at this position and an *template_en* 2102 signal is asserted.

The signature generator 1406 of Fig 14. is shown in greater detail in Fig. 22 and is used to determine an outer and inner ring signature in accordance with the signature detect mask 1000 shown in Fig. 10 and Fig. 11.

The content of the pattern store 1402 is passes through the outer region OR-tree 2200 which is hardwired to look at the shaded cell locations in the signature detect mask 1000 of Fig 10. The outer region OR-tree 2200 takes vertical slices through the pattern store and OR bits found coincident at the shaded cell locations, of the signature detect mask 1000 of

Fig. 10, together via an outer Or-shift register 2201. The output 2202 of the outer OR-shift register 2201 is a 36-bit value representing an outer ring signature for a mark at a current data position in the pattern store 1402. The output 2202 is also directed through an outer FG adder tree 2203 to determine the number mark elements detected for the outer ring signature. In a substantially similar manner a 18-bit inner ring signature 2206 is generated by an inner region OR-tree 2204 and an inner Or-shift register 2205 from the content of the pattern store 1402 in accordance with the signature detect mask 1007 of Fig. 11. However the data from the pattern store 1402 is first pushed through a 9 stage pipeline 2207 to align the data in accordance with the relative positions of signature detect mask 1007 and signature detect mask 1000 of Fig. 11 and Fig 10 respectively. An inner FG adder tree 2208 is used to determine the number mark elements detected for the inner ring signature.

Fig.23 is a block diagram of the signature correlator 1407 of Fig 14 in more detail. The signature correlator 1407 performs a correlation between signatures generated by the signature generator 1406 and predetermined or known signatures for each of the marks represented in Fig. 3 and Fig. 4. In addition the signature correlator 1407 determines a correlation error for each of 36 possible rotations of the signature.

A known outer ring signature for each of the two marks in Fig 3 or Fig 4 is selected by a multiplexer 2300 according to whether a current line is active or active plus one. This is passed to a set of three identical AND networks 2301 that simply gate the known outer ring signature values with an output of a 12-1 multiplexer (MUX) tree 2304. The 12-1 MUX tree 2304 selects three outer signatures according to which *hclk* signal clock period is current relative to an incoming outer signature signal 2202 (from the signature generator Fig 22). The known outer ring signature values first passes through a MAP function 2305 that maps values from the known outer ring signature to real weighting values as defined in template weighting registers. The outputs from the AND networks 2301 passes to a set of three identical adder trees 2306 which produce a sum corresponding to a correlation error between the known outer ring signature and the incoming outer signature signal 2202. A substantially similar circuit produces correlation error values between a known inner ring signature for each of the two marks in Fig 3 or Fig 4 and an incoming inner signature signal 2206 (also shown in Fig 22). The inner correlation values are added 2307 to the outer correlation error values to make a total correlation errors for each signature. The total correlation error values go into a first minimum block unit 2308 which passes on a smallest of three input values to a pipeline register 2309 that is used to hold the total correlation error values.

The total correlation values are compared, by a second minimum block unit 2310, to each other to determine a lowest value, which is passed on to a compatibility calculator 1409 (Fig. 14) via a correlation error bus 2311 (corr_err signal). The lowest total correlation value is used to determine a confidence level or 'goodness of fit' of a match to one or other of the marks shown in Fig. 3 and Fig.4.

Fig 24 shows the compatibility calculator 1409 in Fig 14 in greater detail. The compatibility calculator comprises a set of range comparators 2400 that compare an incoming corr_err value (i.e. lowest total correlation value output from the signature correlator 1407) to each of a plurality of predetermined thresholds 2401. The output of the range comparators 2400 is a set of 3 range detects that immediately pass into an encoder 2402 which encodes the range detect input into a 2-bit value corresponding to an input corr_err value range that was determined. This 2-bit range value selects one of four possible range values to pass to the evidence combiner module 1302, via a pc_compat bus 2403.

The lowest total correlation value is thus split into four regions according to programmable thresholds values, which are then assigned to one of four fixed values in compatibility space. The four pre-defined fixed values are programmed via a plurality of range values 2404 and which preferably have a value set between 0 and 100.

The evidence combiner module 1302 takes the value of compatibility and enables or disables functions of an input device, output device or a combined input/output device such as a photocopier in accordance with a thresholding criteria on the compatibility value. For example the functions may include the prevention of copying, the sounding of a warning or the shutting down of the device in which the present invention is installed.

The foregoing describes a preferred embodiment of the present invention, and modifications, obvious to those skilled in the art can be made thereto without departing from the spirit and scope of the present invention.

Aspects of the Invention

The following numbered paragraphs set forth aspects of the invention:

1. An apparatus for sharpening digital data comprising:

an array of logic cells, each cell being individually configurable to provide a plurality of functions;

a pipeline register, having a plurality of input means for receiving said digital data and for pipelining said data through said array of logic cells;

a plurality of logic gates for detecting a desired pattern of digital data, and for enabling said logic cells to perform said configured functions on said digital data.

2. An apparatus of paragraph 1, wherein the desired pattern is provided by said function configured logic cells.

3. An apparatus of paragraph 1, wherein said pipeline register comprises a plurality of flip-flop circuits in a matrix (array) arrangement which forms a pipeline shift register.

4. An apparatus of paragraph 3, wherein said logic gates clears selected flip-flops via said logic cells when a predetermined pattern is detected.

5. An apparatus of paragraph 4, wherein said predetermined pattern determined by the configuration functions and the arrangement of the flip-flop array and logic cells.

DATED this THIRTIETH day of NOVEMBER 1998

Canon Kabushiki Kaisha

Patent Attorneys for the Applicant

SPRUSON & FERGUSON

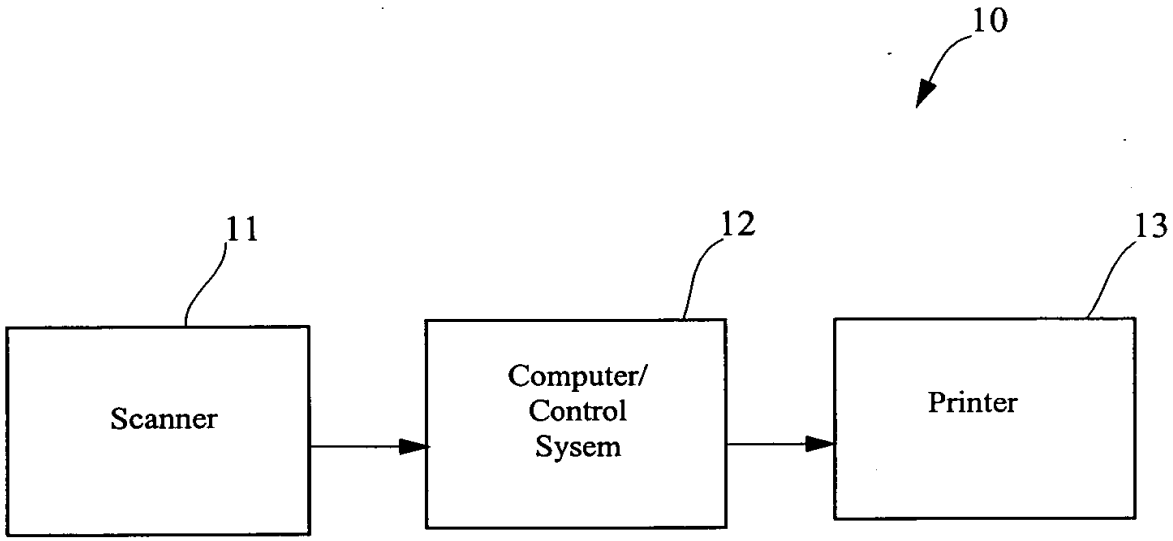

Fig. 1

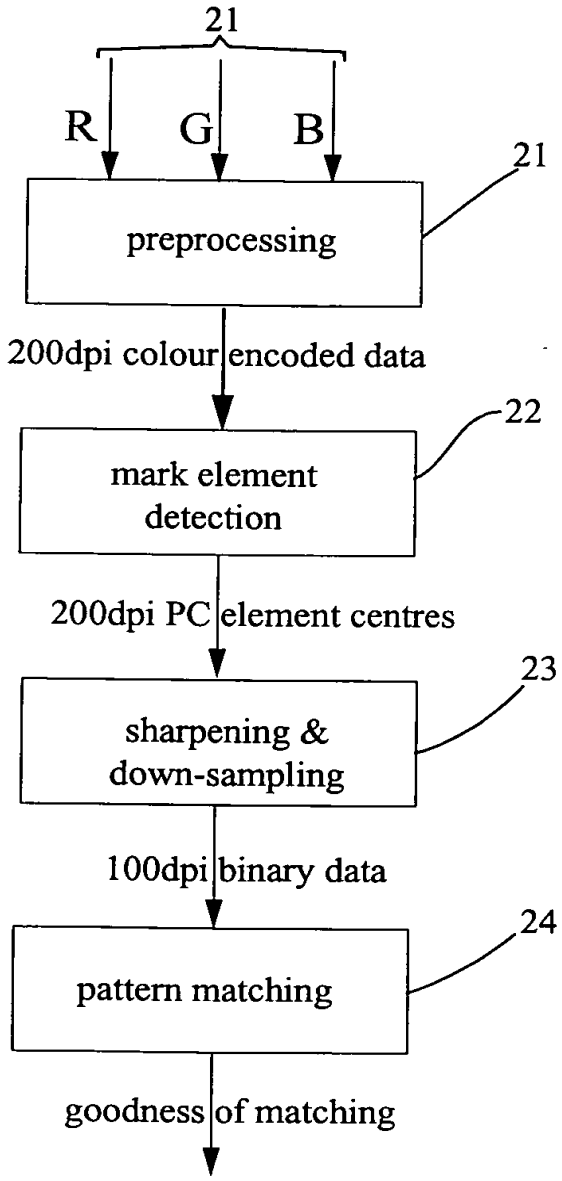
21
R
G
B
21
preprocessing
200dpi colour encoded data
22
mark element detection
200dpi PC element centres
23
sharpening & down-sampling
100dpi binary data
24
pattern matching
goodness of matching

Fig. 2

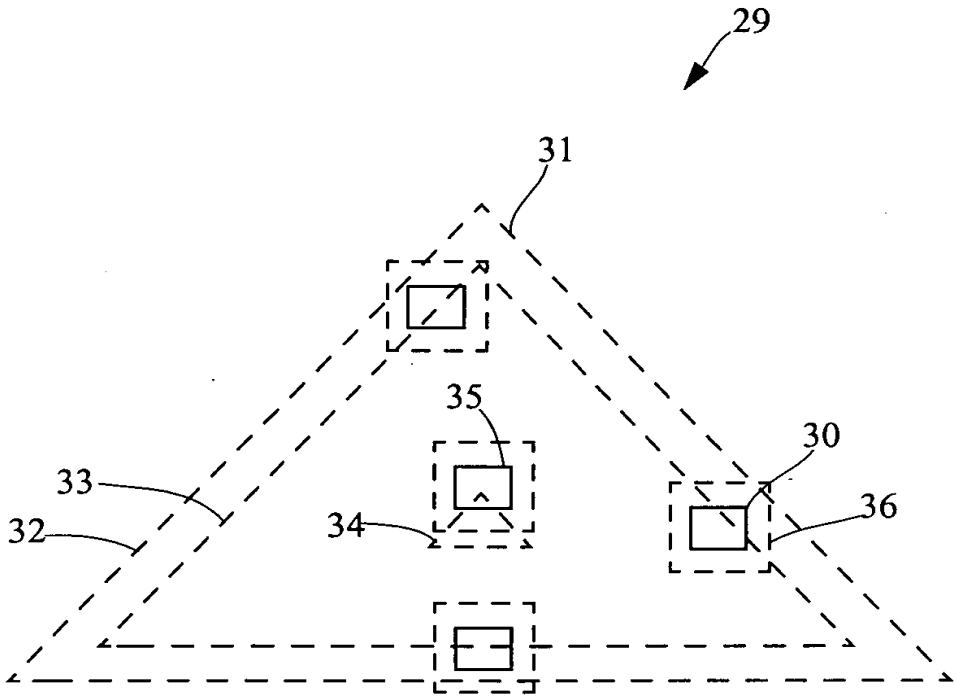

Fig. 3

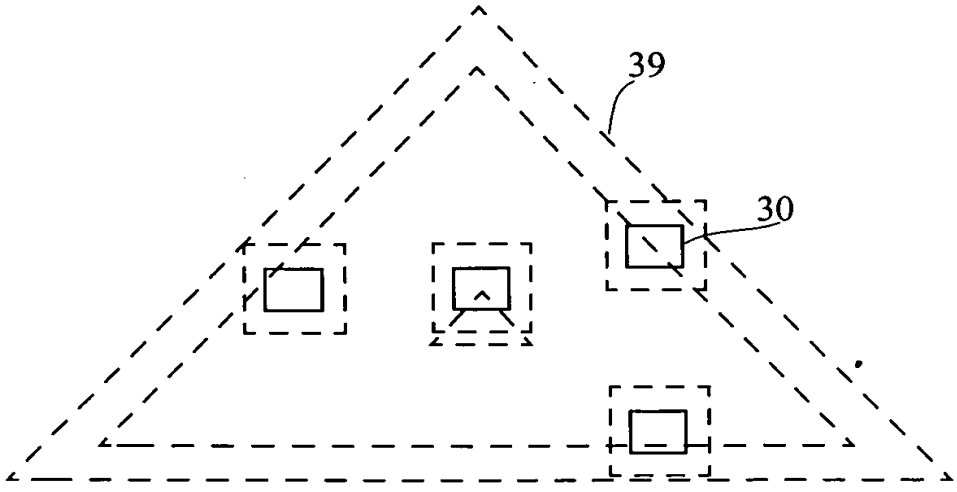

Fig. 4

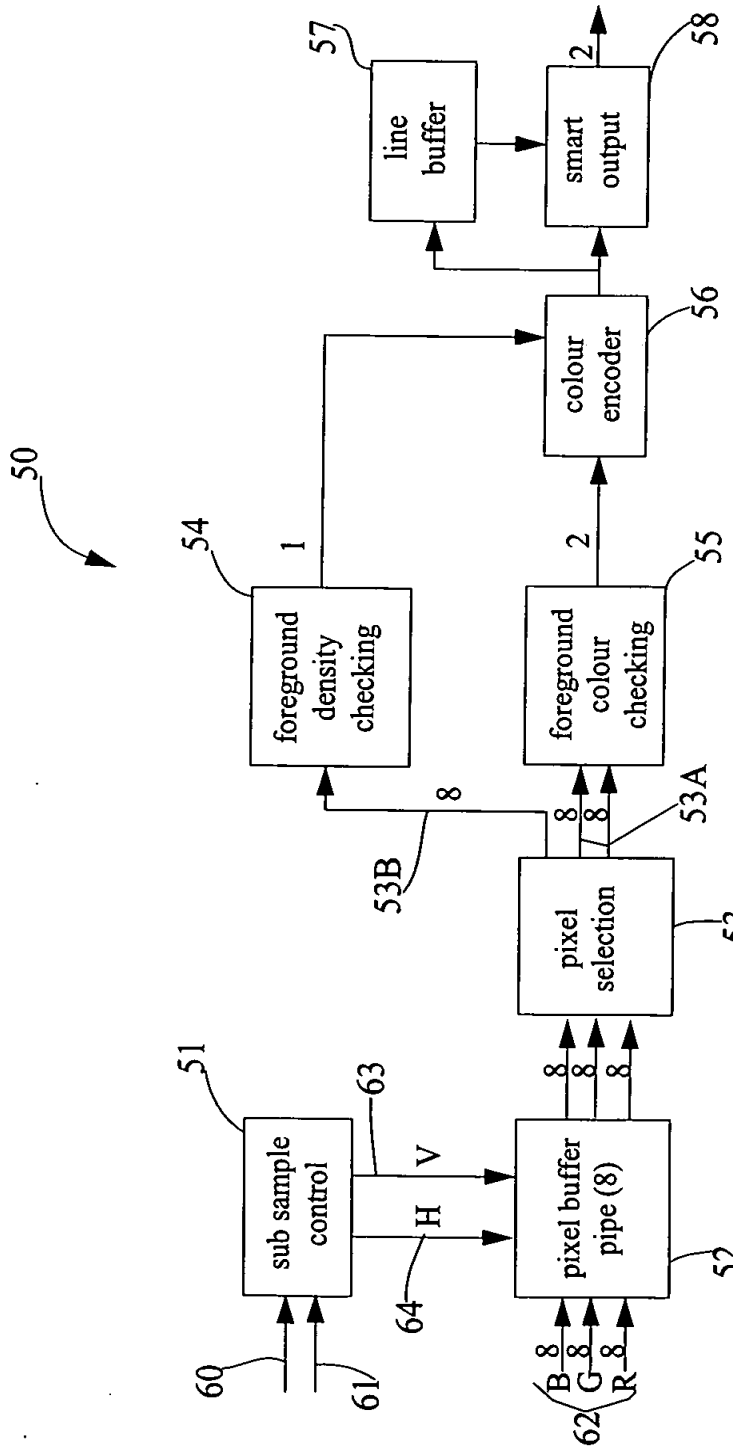
50
51
sub sample control
63
V
H
64
60
61
52
pixel buffer pipe (8)
8
8
8
B
G
R
62
53
pixel selection
53A
53B
54
foreground density checking
1
55
foreground colour checking
2
56
colour encoder
57
line buffer
58
smart output
2

Fig. 5

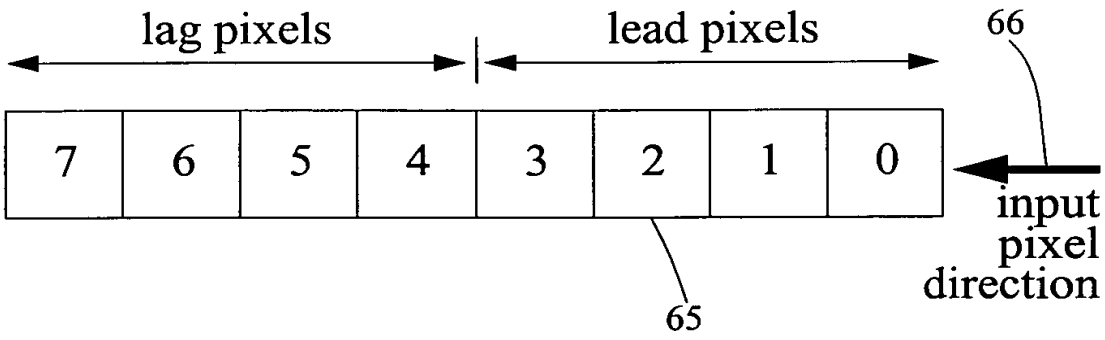
lag pixels
lead pixels
66
7
6
5
4
3
2
1
0
input
pixel
direction
65

Fig. 6

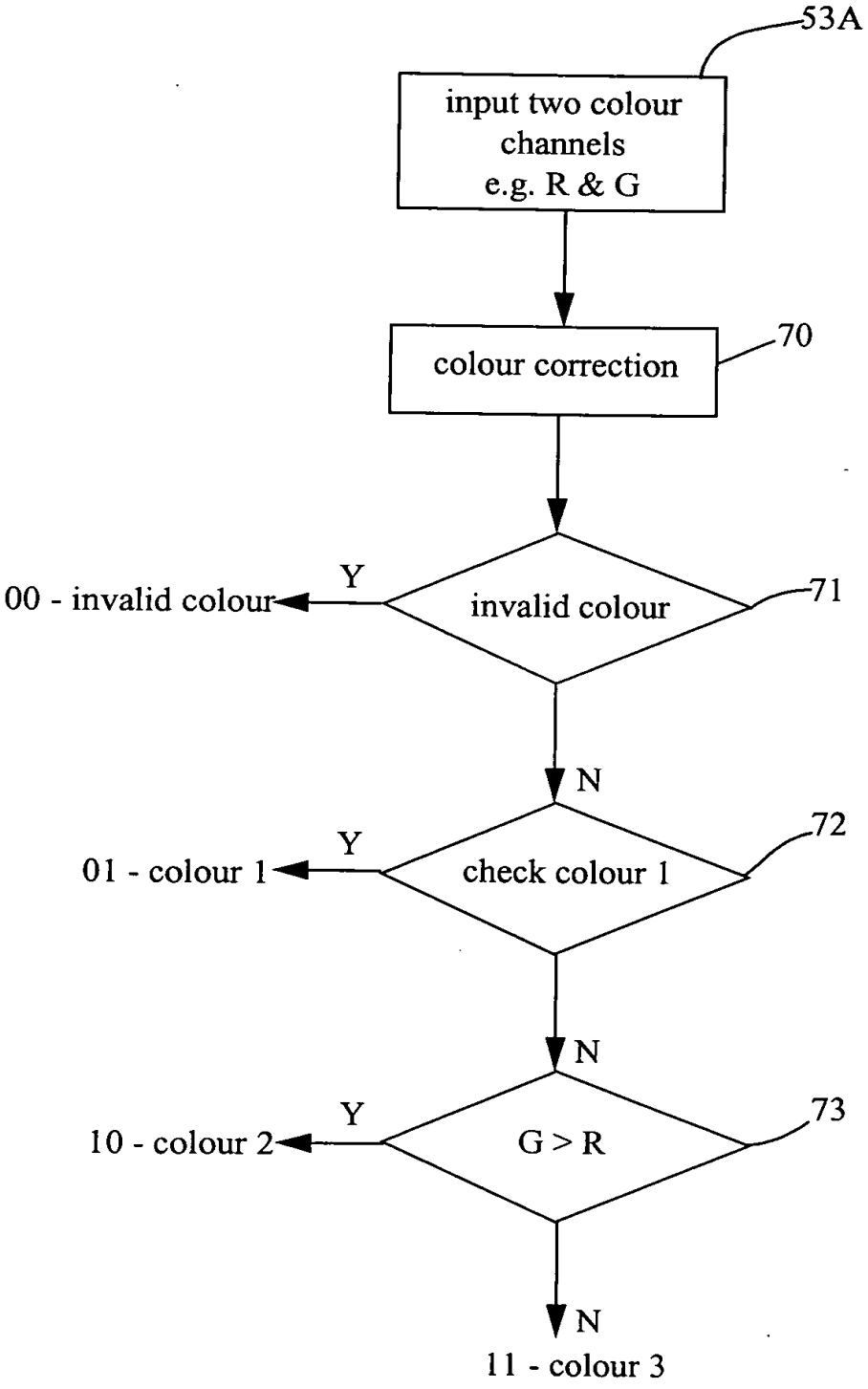
53A
input two colour channels e.g. R & G
colour correction
70
invalid colour
71
Y
00 - invalid colour
N
check colour 1
72
Y
01 - colour 1
N
G > R
73
Y
10 - colour 2
N
11 - colour 3

Fig. 7

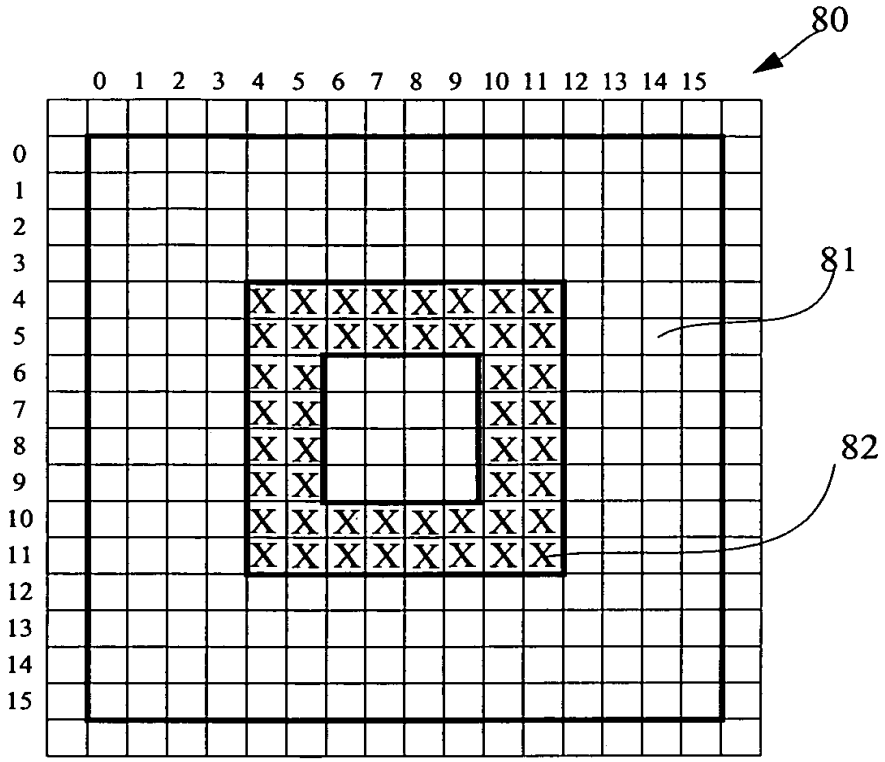
80
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
81
82

Fig. 8

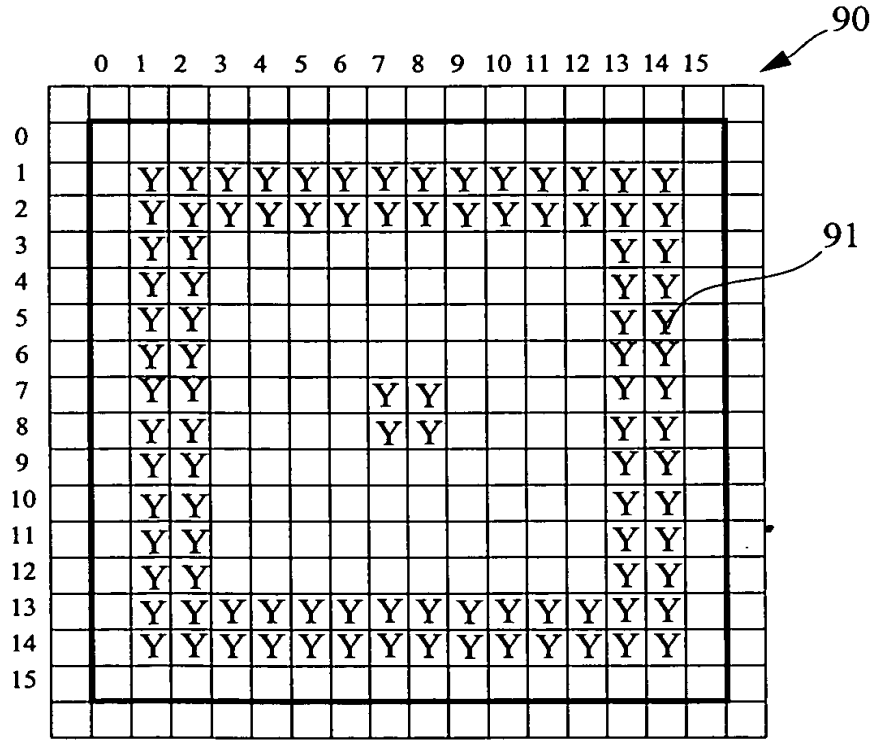
90
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
91

Fig. 9

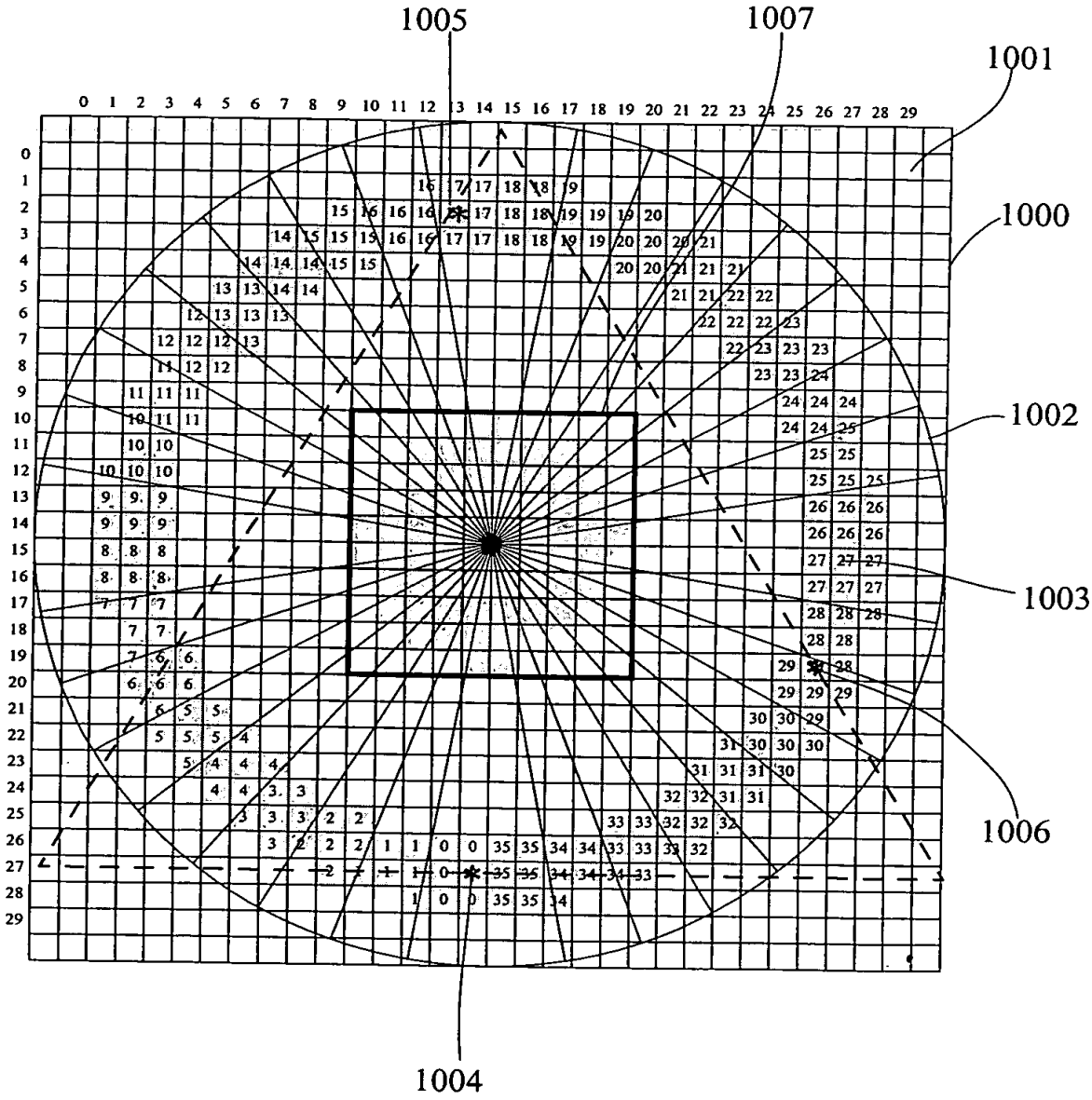

Fig. 10

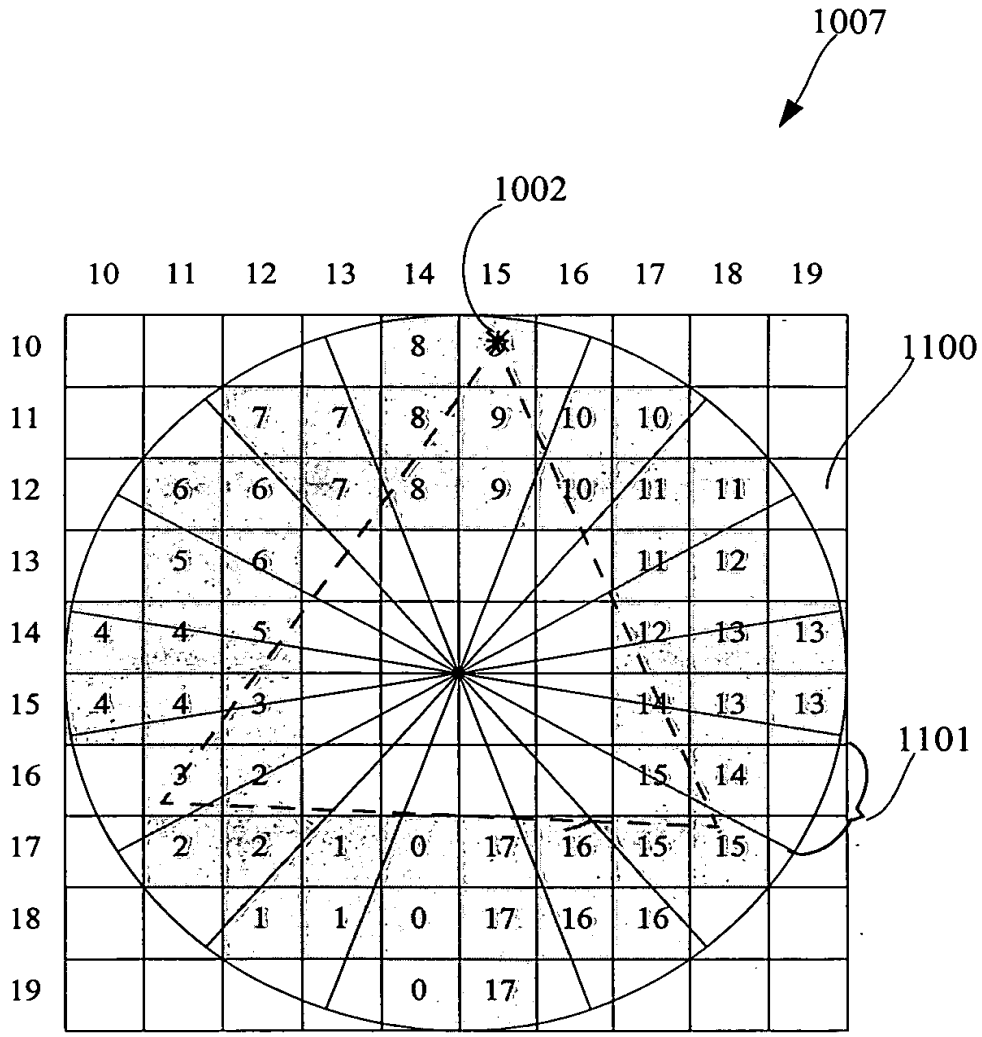
1007
1002
1100
1101
10 11 12 13 14 15 16 17 18 19
10 8
11 7 7 8 9 10 10
12 6 6 7 8 9 10 11 11
13 5 6 11 12
14 4 4 5 12 13 13
15 4 4 3 14 13 13
16 3 2 15 14
17 2 2 1 0 17 16 15 15
18 1 1 0 17 16 16
19 0 17

Fig. 11

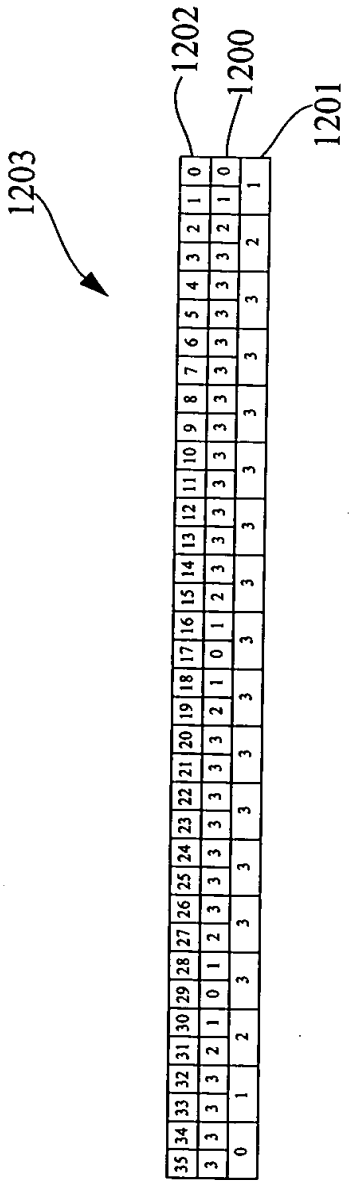

Fig. 12

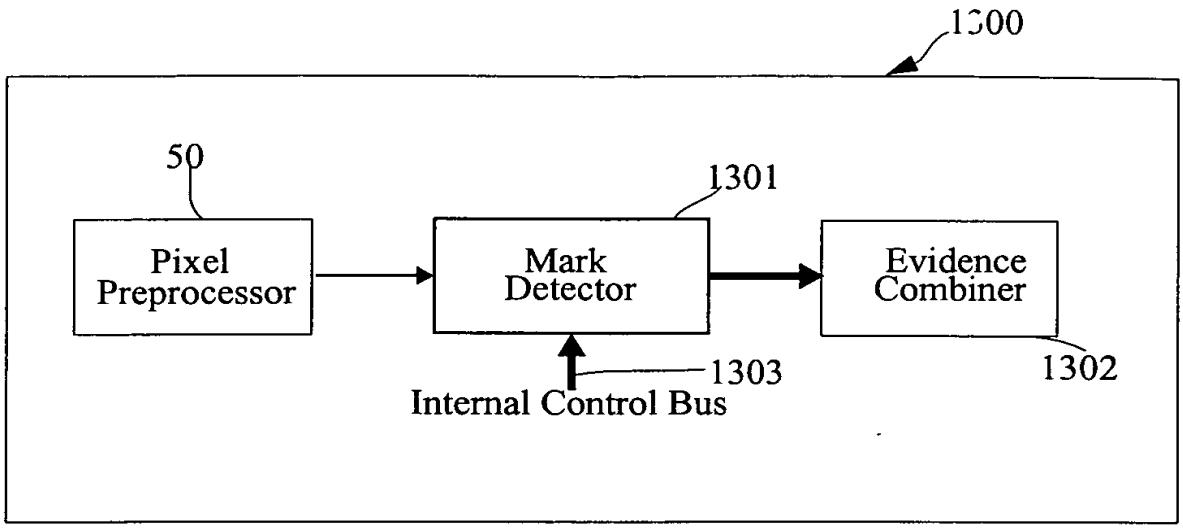
1300
50
1301
Pixel Preprocessor
Mark Detector
Evidence Combiner
1303
1302
Internal Control Bus

Fig. 13

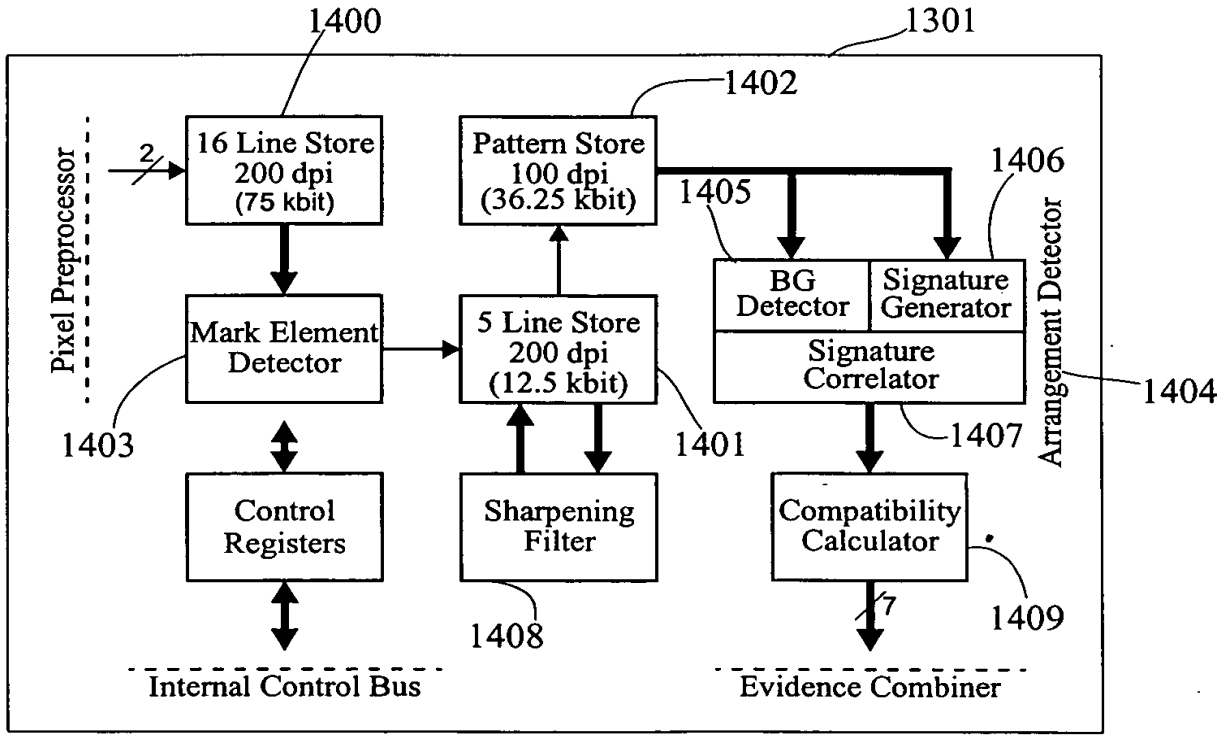
1400
1301
1402
16 Line Store 200 dpi (75 kbit)
Pattern Store 100 dpi (36.25 kbit)
1406
1405
2
Pixel Preprocessor
BG Detector
Signature Generator
Arrangement Detector
Mark Element Detector
5 Line Store 200 dpi (12.5 kbit)
Signature Correlator
1404
1403
1401
1407
Control Registers
Sharpening Filter
Compatibility Calculator
1409
7
1408
Internal Control Bus
Evidence Combiner

Fig. 14

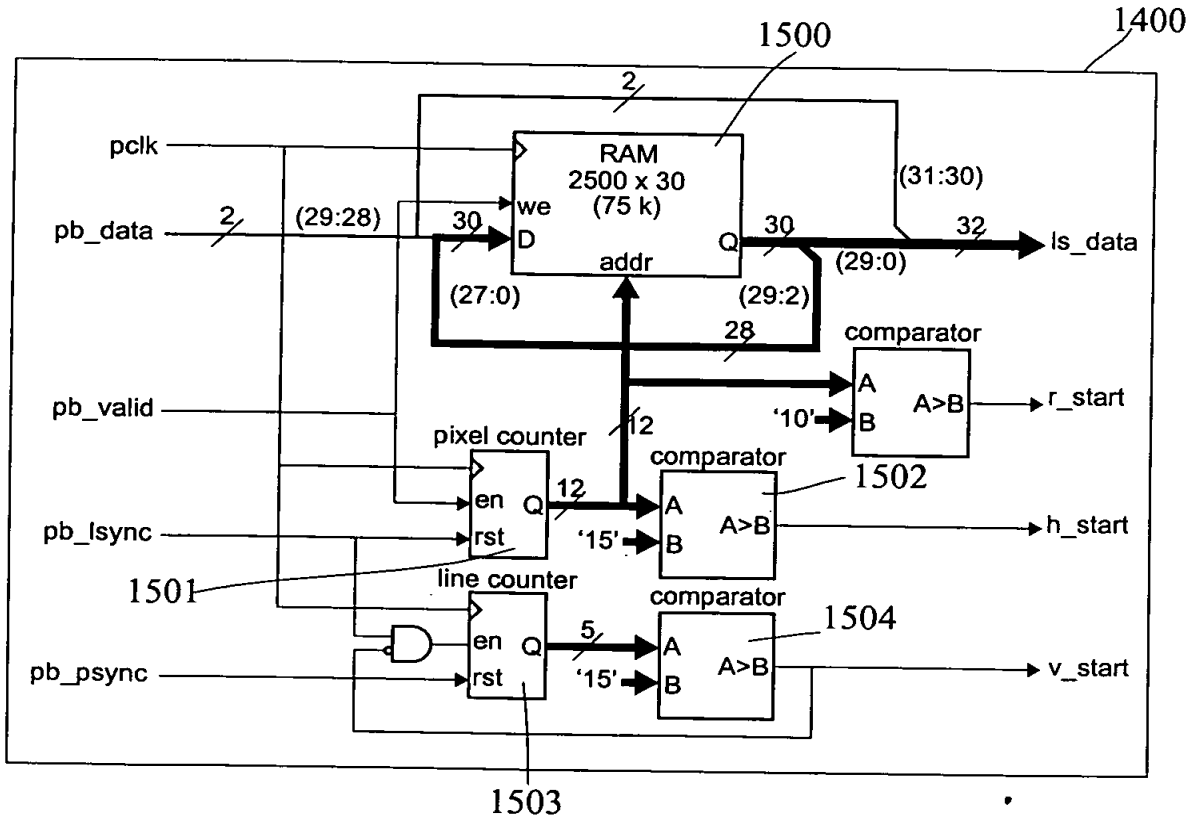
1400
1500
2
pclk
RAM
2500 x 30
(75 k)
we
D
addr
Q
(31:30)
pb_data
2
(29:28)
30
30
32
ls_data
(29:0)
(27:0)
(29:2)
28
comparator
A
B
A>B
r_start
'10'
pb_valid
pixel counter
12
comparator
1502
en
Q
rst
A
B
A>B
h_start
'15'
pb_lsync
1501
line counter
comparator
1504
en
Q
5
rst
A
B
A>B
v_start
'15'
pb_psync
1503

Fig. 15

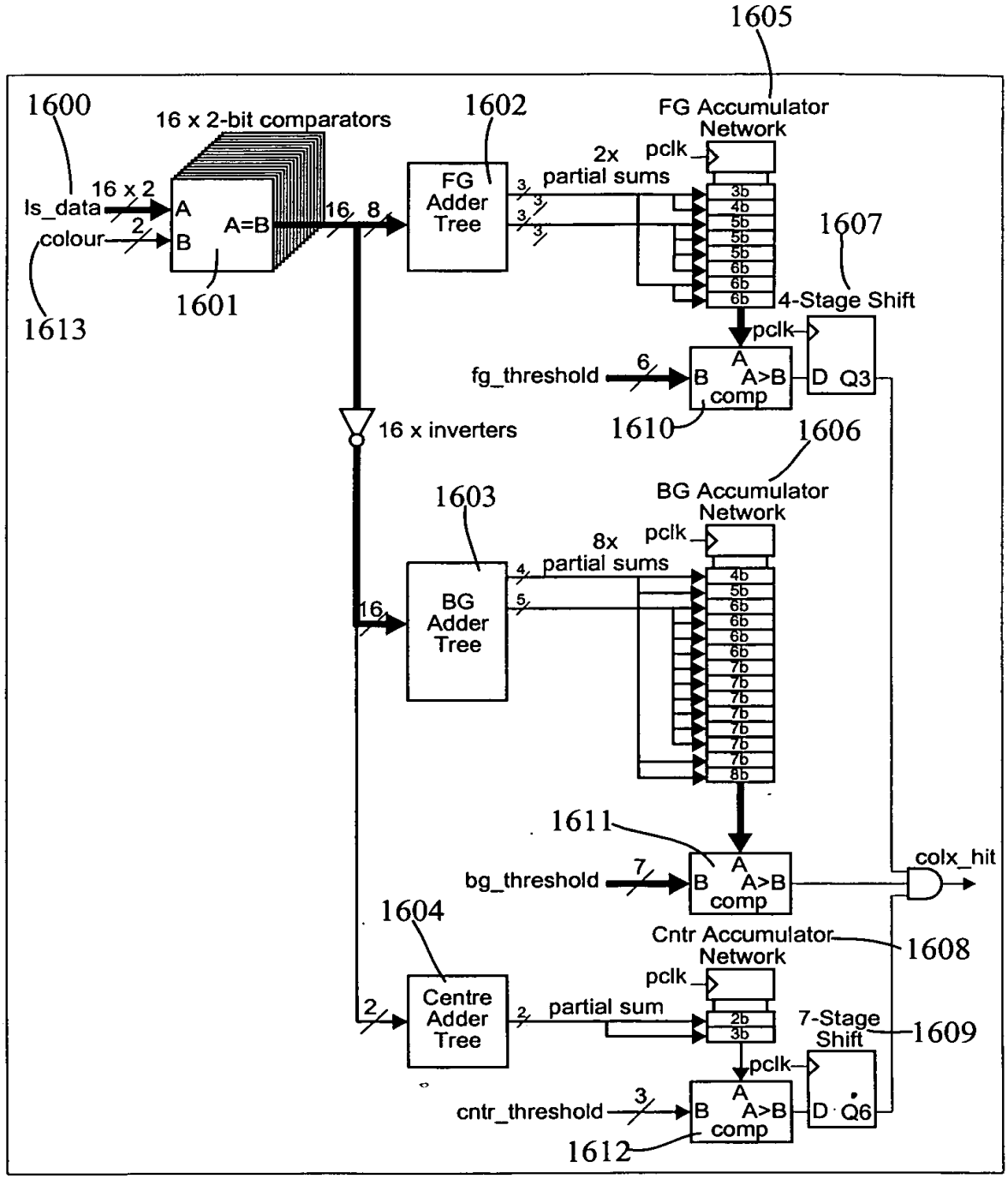
1605
1600
16 x 2-bit comparators
1602
FG Accumulator Network
ls_data
16 x 2
A
B
A=B
16
8
FG Adder Tree
2x
pclk
partial sums
3b
4b
5b
5b
5b
6b
6b
6b
colour
2
1613
1601
1607
4-Stage Shift
pclk
fg_threshold
6
A
B
A>B
comp
D Q3
1610
16 x inverters
1606
1603
BG Accumulator Network
8x
pclk
partial sums
4b
5b
6b
6b
6b
6b
7b
7b
7b
7b
7b
7b
7b
8b
16
BG Adder Tree
1611
bg_threshold
7
colx_hit
1604
Cntr Accumulator Network
1608
pclk
Centre Adder Tree
2
partial sum
2b
3b
7-Stage Shift
1609
pclk
cntr_threshold
3
D Q6
1612

Fig. 16

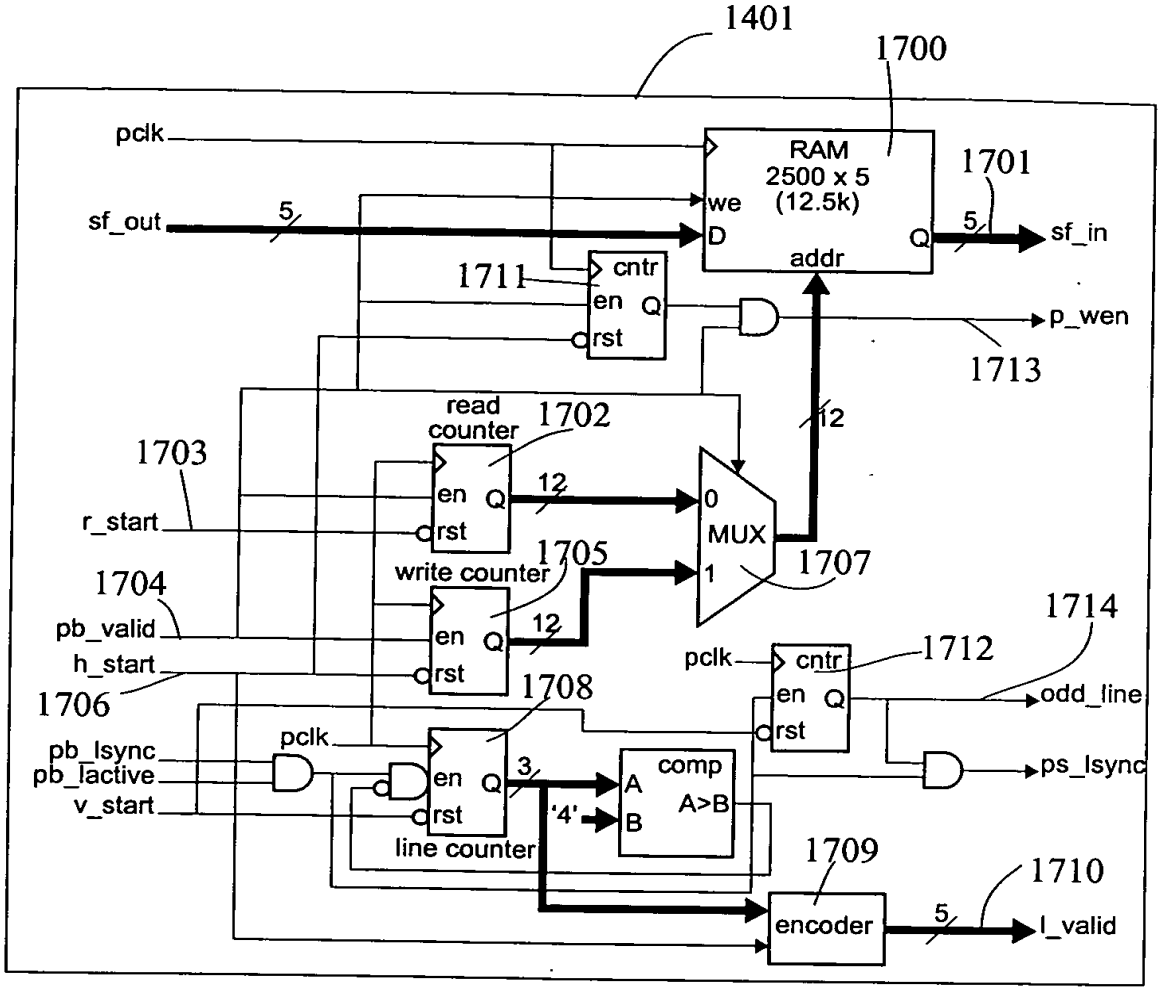

Fig. 17

1800

0	0	0	0	0
0	X	X	X	0
0	X	1	X	0
0	X	X	X	0
0	0	0	0	0

Fig. 18A

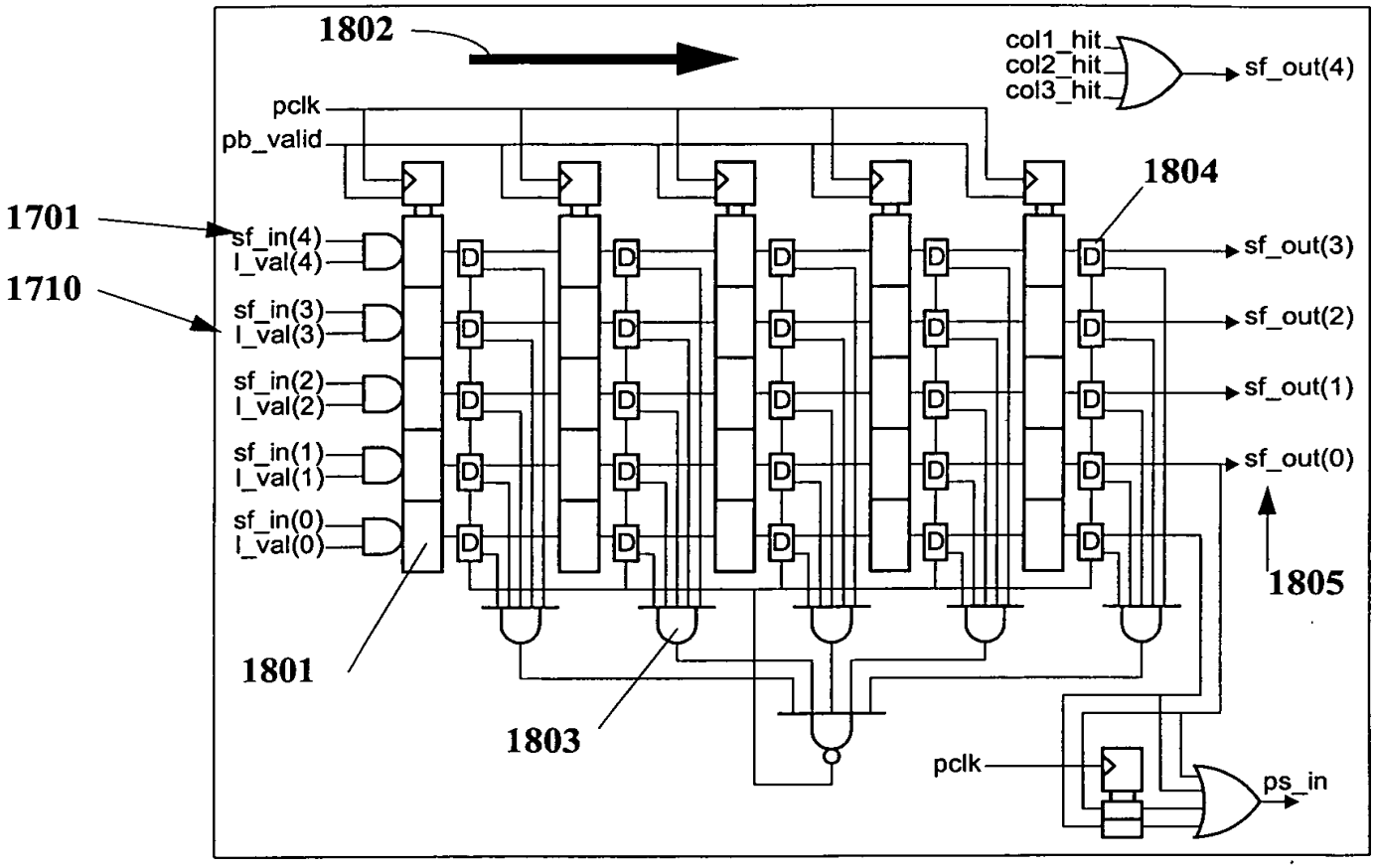

Fig 18 B

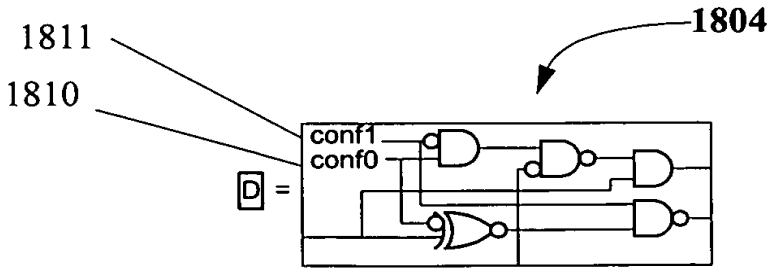

Fig 18C

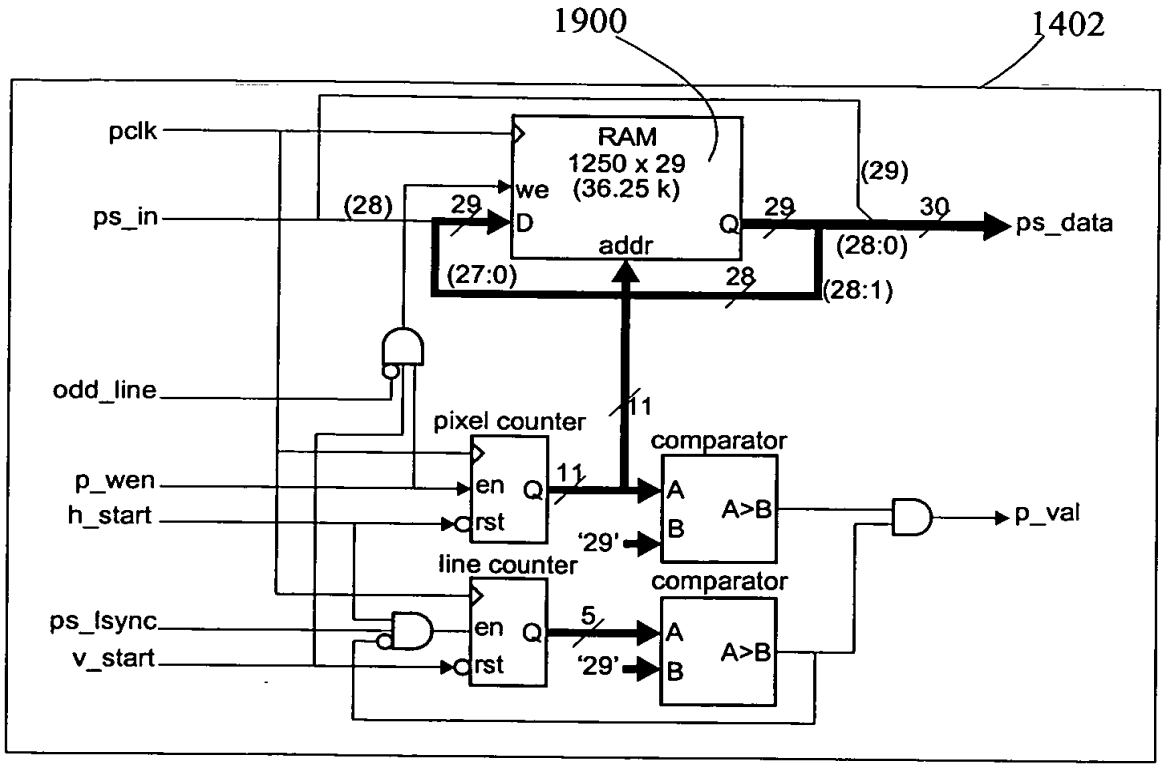
1900
1402
pclk
ps_in
RAM
1250 x 29
(36.25 k)
we
D
Q
addr
(28)
29
(27:0)
28
(28:1)
29
(29)
30
(28:0)
ps_data
odd_line
pixel counter
11
11
en
rst
comparator
A
B
A>B
'29'
p_wen
h_start
p_val
line counter
5
'29'
ps_fsync
v_start

Fig. 19

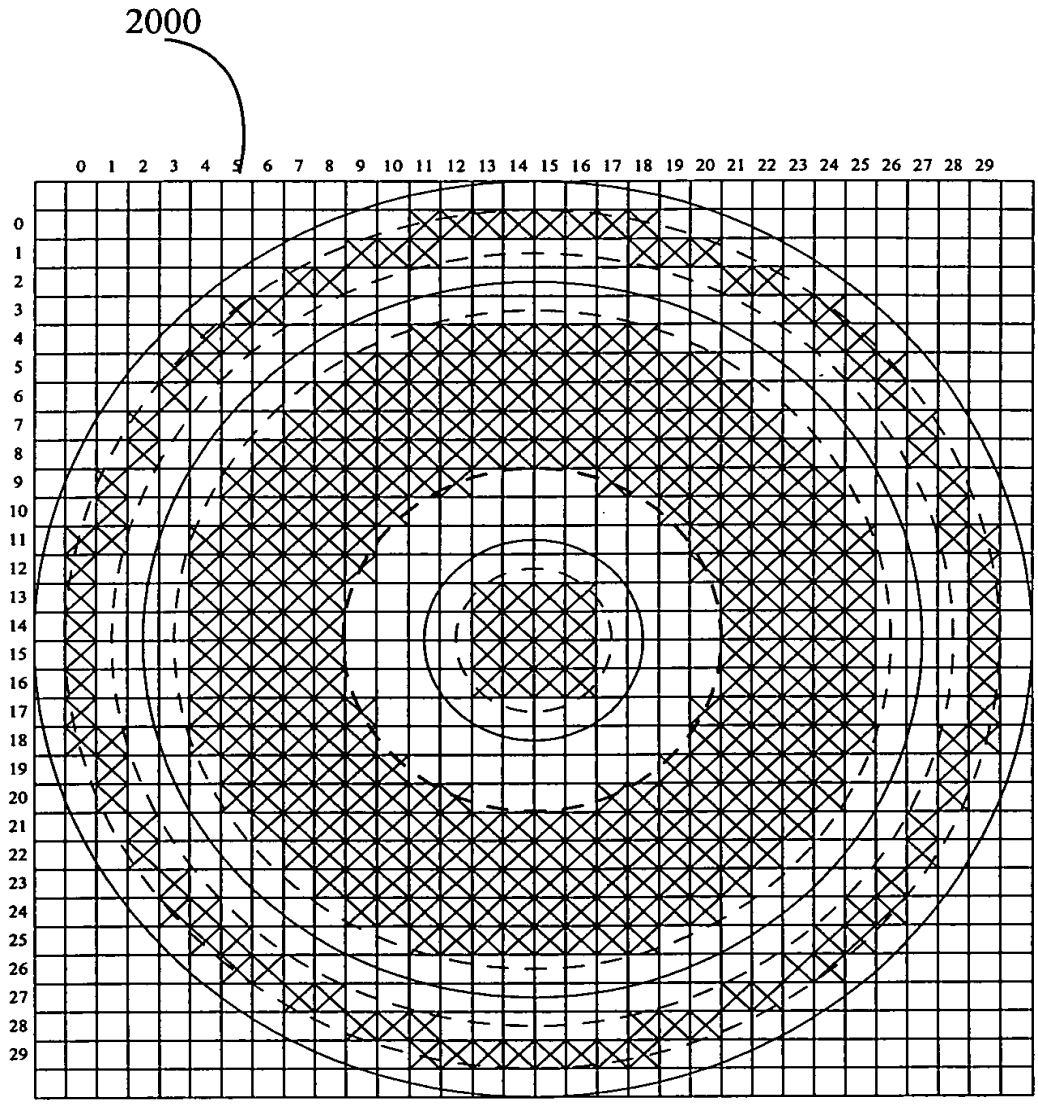
2000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

Fig.20

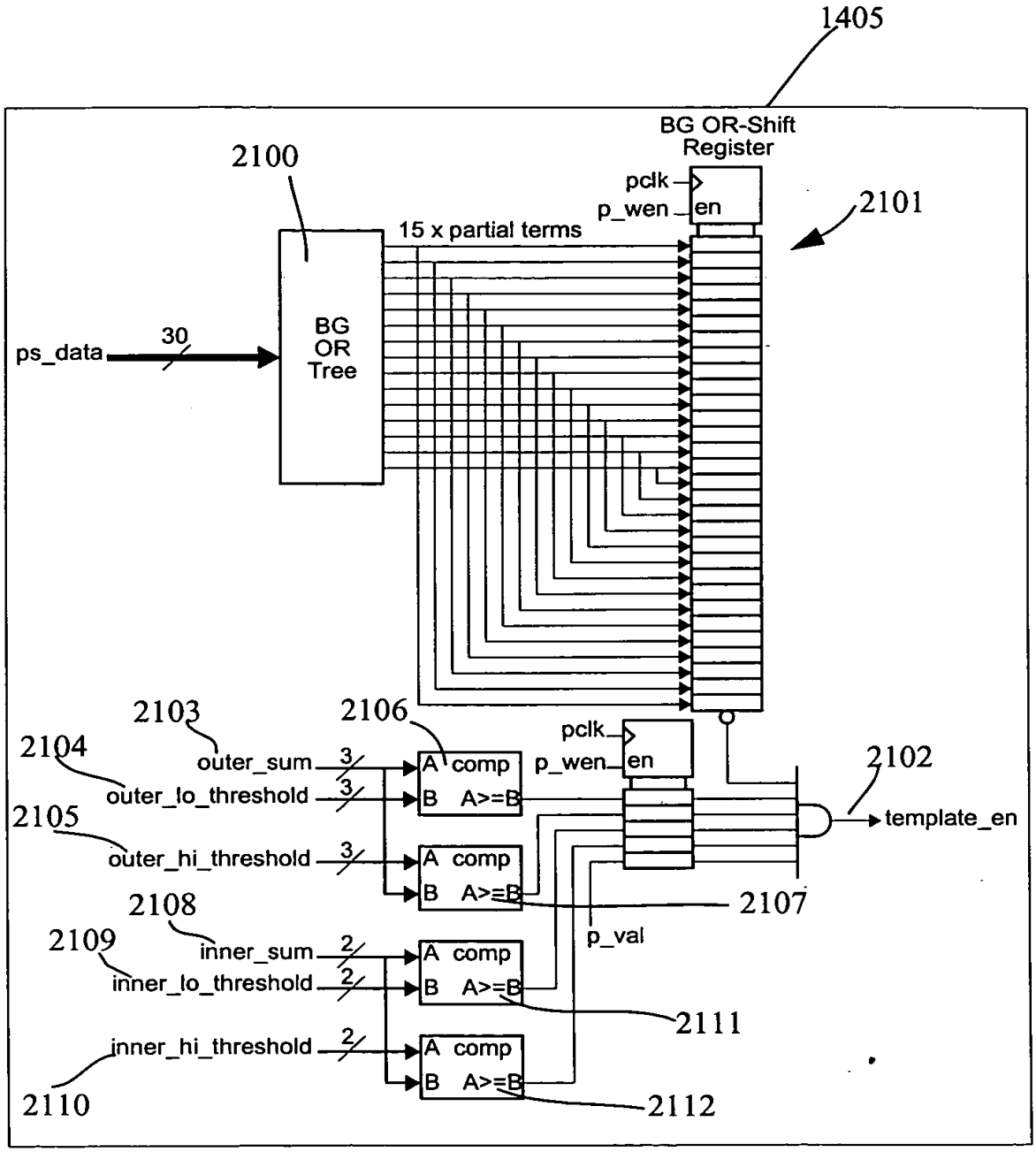
1405
BG OR-Shift Register
2100
pclk
p_wen
en
2101
15 x partial terms
ps_data
30
BG OR Tree
2103
2106
2104
outer_sum
3
A comp
outer_lo_threshold
3
B A>=B
pclk
p_wen
en
2102
2105
outer_hi_threshold
3
A comp
B A>=B
template_en
2107
2108
p_val
2109
inner_sum
2
A comp
inner_lo_threshold
2
B A>=B
2111
inner_hi_threshold
2
A comp
2110
B A>=B
2112

Fig.21

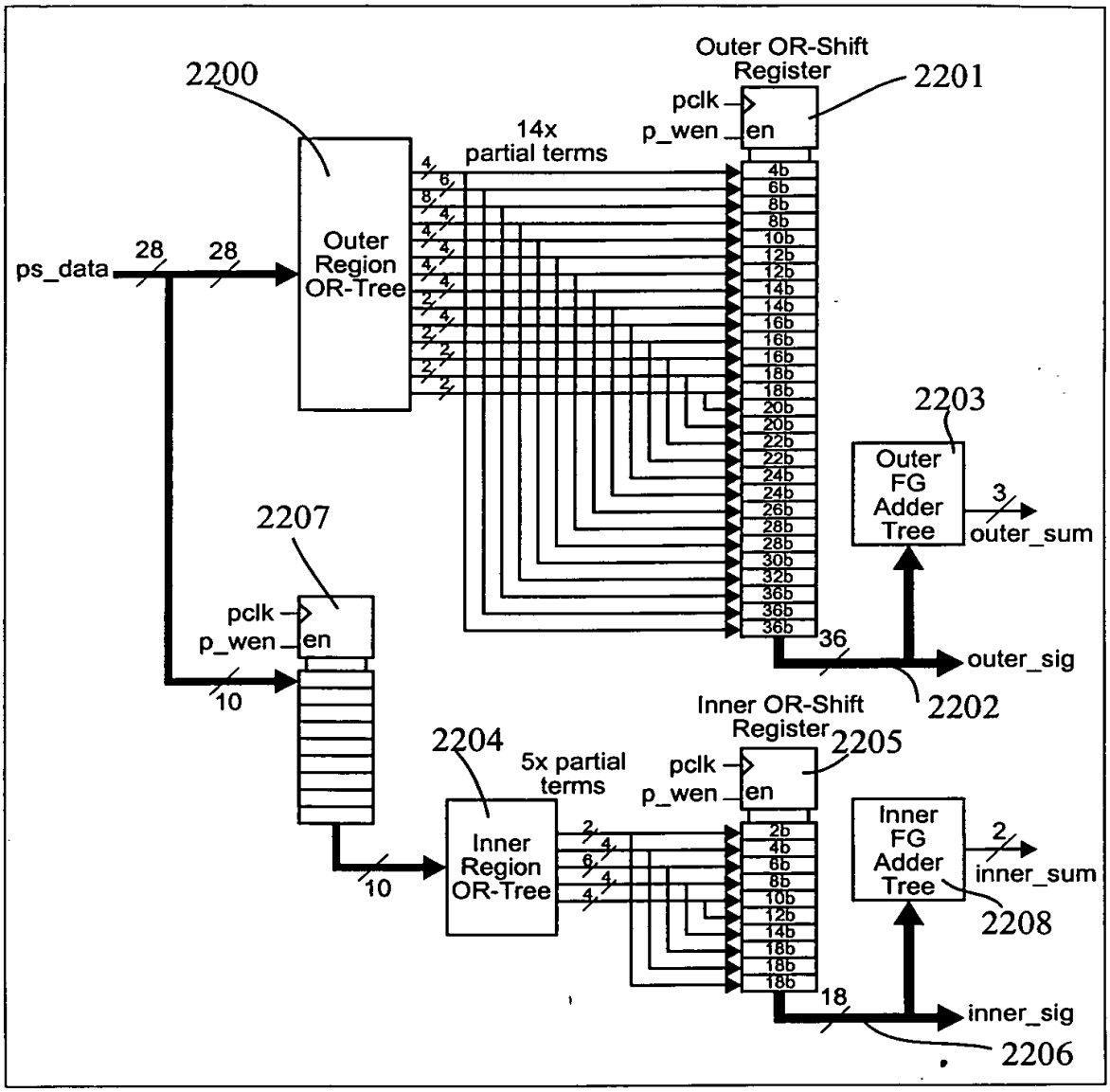
2200
Outer OR-Shift Register
2201
pclk
p_wen
en
14x partial terms
ps_data
28
28
Outer Region OR-Tree
4b
6b
8b
8b
10b
12b
12b
14b
14b
16b
16b
16b
18b
18b
20b
20b
22b
22b
24b
24b
26b
28b
28b
30b
32b
36b
36b
36b
2203
Outer FG Adder Tree
3
outer_sum
36
outer_sig
2202
2207
pclk
p_wen
en
10
Inner OR-Shift Register
2205
2204
5x partial terms
pclk
p_wen
en
Inner Region OR-Tree
10
2b
4b
6b
8b
10b
12b
14b
18b
18b
18b
Inner FG Adder Tree
2
inner_sum
2208
18
inner_sig
2206

Fig.22

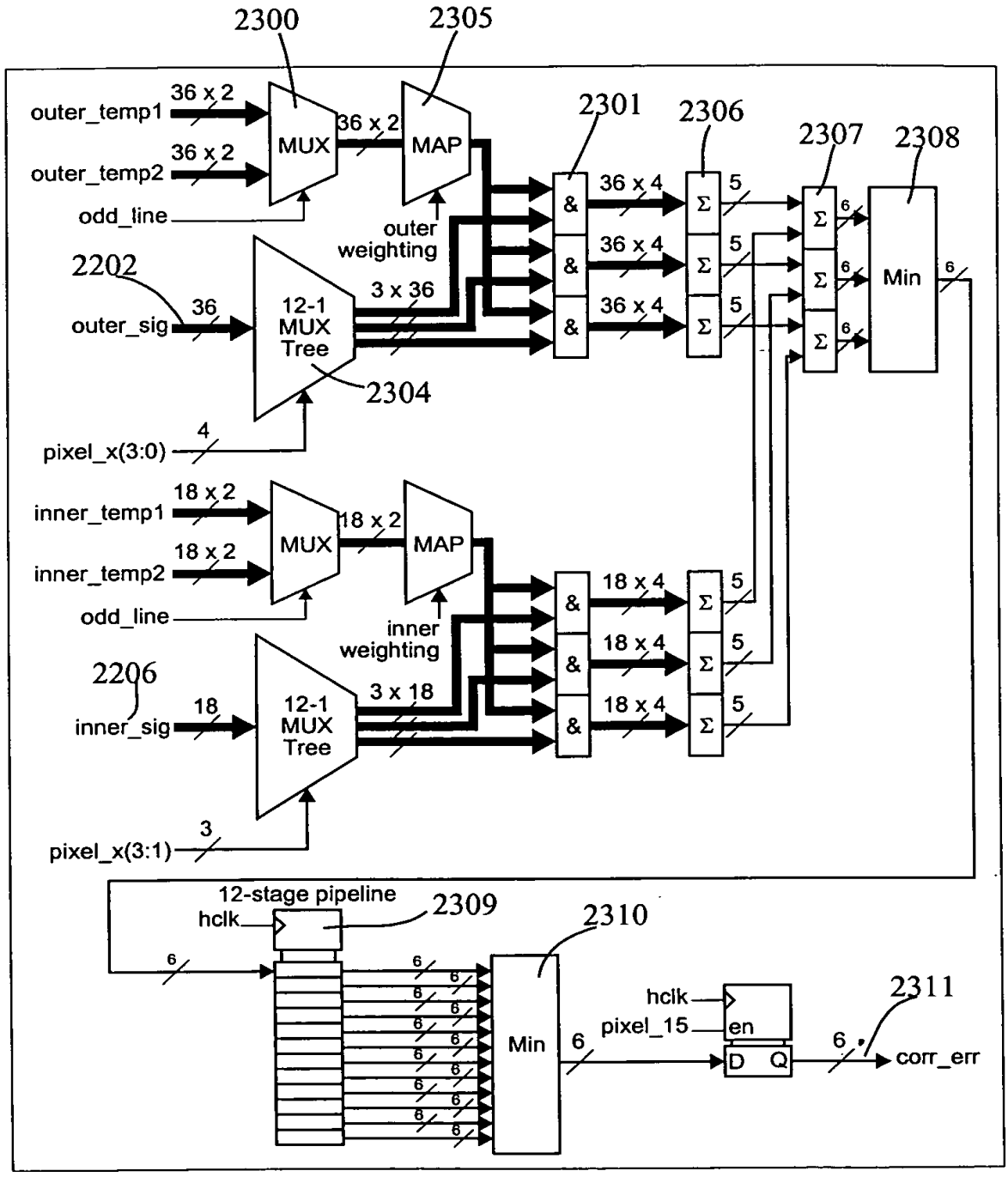
2300
2305
outer_temp1
36 x 2
outer_temp2
36 x 2
MUX
36 x 2
MAP
odd_line
outer weighting
2301
2306
2307
2308
36 x 4
5
Σ
6
Min
2202
outer_sig
36
12-1 MUX Tree
3 x 36
2304
pixel_x(3:0)
4
inner_temp1
18 x 2
inner_temp2
18 x 2
MUX
18 x 2
MAP
odd_line
inner weighting
18 x 4
2206
inner_sig
18
12-1 MUX Tree
3 x 18
pixel_x(3:1)
3
12-stage pipeline
hclk
2309
2310
Min
hclk
pixel_15
en
D Q
2311
corr_err

Fig.23

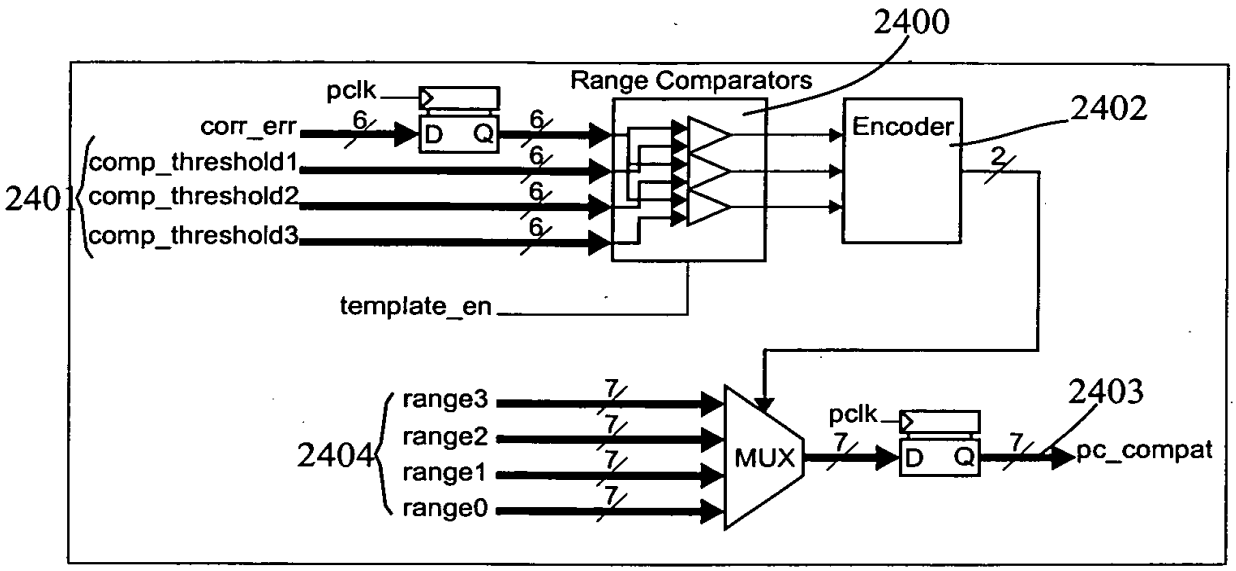
2400
Range Comparators
pclk
corr_err
6
D Q
6
comp_threshold1
comp_threshold2
comp_threshold3
240
Encoder
2402
2
template_en
range3
range2
range1
range0
2404
7
MUX
pclk
2403
D Q
pc_compat

Fig.24